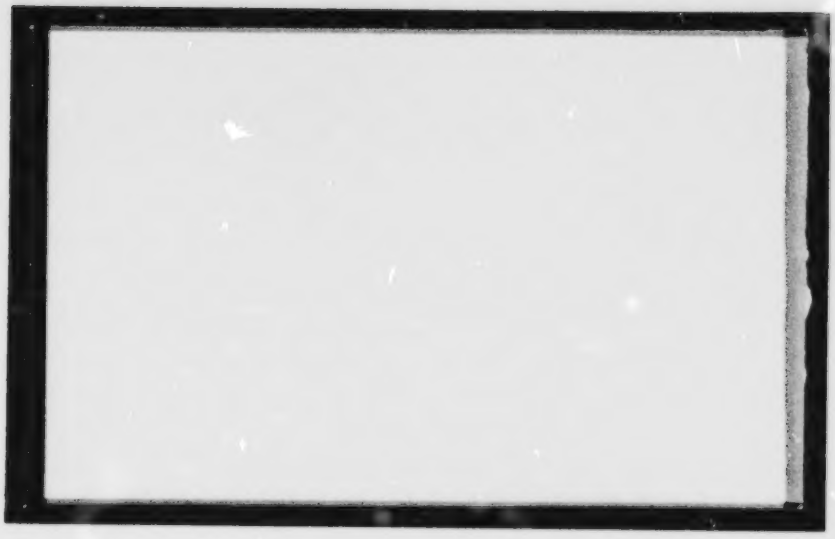
UNIVERSITÉ DE MONTRÉAL



CENTRE DE RECHERCHE SUR LES TRANSPORTS

MÉMOIRE D'ÉTUDIANT

CASE POSTALE 6128, STATION A, MONTRÉAL, QUÉBEC, H3C 3J7

GENERATION DES PARCOURS POUR LE TRANSPORT SCOLAIRE EN MILIEU RURAL

par
Nada Tawagui

Ce travail a été subventionné par le Centre de recherche et de développement de Transports Canada et par le programme F.C.A.C. du Ministère de l'Education du Québec

Université de Montréal

Centre de recherche sur les transports - Publication # 159

Août 1979

Dépôt légal - 1er trimestre 1980

SOMMAIRE

Ce document traite de la génération de parcours d'autobus pour le transport scolaire en milieu rural. Les heuristiques employés pour résoudre le problème en milieu urbain (Forgues [12]) y ont été adaptés. Les heuristiques de Clarke & Wright [16], Insertion [15] et Yellow [24], tel qu'adaptés au problème de génération de parcours sur les segments sont décrits et les détails propres au milieu rural mentionnés. Des programmes ont été développés pour chaque heuristique; ils ont permis une comparaison des performances des heuristiques exécutés sur huit problèmes. Les problèmes de l'affectation des parcours aux points de transfert pour les étudiants de secondaire et de réduction du temps moyen qu'un étudiant passe dans l'autobus sont aussi traités.

REMERCIEMENTS

Je désire remercier tous les professeurs du Département d'Informatique et de Recherche Opérationnelle qui m'ont aidée par leurs conseils et leur enseignement. Je remercie en particulier mon directeur de stage, le Professeur Jacques Ferland, pour son assistance et ses conseils. Je remercie également, le Professeur Jean-Marc Rousseau ainsi que Mr. Guy Lapalme pour leur disponibilité leur support technique et méthodologique.

Je remercie la Commission Scolaire Régionale St-François en Monsieur Yvon Levesque, régisseur au transport scolaire, pour nous avoir procuré les données nécessaires à la génération des parcours et pour ses critiques constructives des résultats.

TABLE DES MATIERES

Chapitre		Page
1.	Introduction	1
2.	Méthodologie	3
3.	Etapes de la planification du transport scolaire	11
4.	Méthode de Clarke & Wright	17
5.	Méthode d'insertion	27
6.	Méthode de Yellow	34
7.	Algorithme d'amélioration de parcours	37
8.	Analyse des résultats	41
9.	Implantation de la génération de parcours en milieu rural	51
10.	Conclusion	64

Annexe

I	Liste sommaire des programmes développés	69

TABLE DES PRINCIPALES ILLUSTRATIONS

Illustrations		Page
2.1	Réseau initial G(N,A) avec les segments et les sommets à parcourir en noir foncé	6
2.2	Création des ensembles A', N'	7
2.3	Création du réseau réduit G(N', A')	7
3.1	Etapes logiques de la génération de parcours en transport scolaire	12
8.1	Description des problèmes traités	42
8.2	Description des heuristiques et paramètres utilisés	44
8.3	Sommaire des solutions heuristiques	45 - 46
8.4	Tableau des solutions choisies	48
8.5	Tableau des deux meilleurs heuristiques	49
9.1	Affectation des segments aux points de transfert pour l'approche II, "diviser d'abord, circuler ensuite"	56
9.2	Comparaison des deux approches	57
9.3	Variation de la distance de ramassage	63

1. Introduction au problème

Le problème de la génération de parcours dans le transport scolaire fait partie d'une étude à long terme comprenant le choix des arrêts et l'assignation des étudiants à ces arrêts (Dulac [8]), la génération des parcours ainsi que l'agencement des parcours dans le temps (Desrosiers [7]).

La génération de parcours est elle-même divisée en deux problèmes distincts : le problème sur les arrêts en milieu urbain, tel que traité par Forgues [12], et le problème plus général sur les arrêts et les segments en milieu rural que nous traitons dans la présente étude.

Nous retrouvons dans le texte une discussion des particularités du problème en milieu rural québécois, et la description détaillée des heuristiques adaptés à ce problème : Clarke & Wright [6], Insertion [15] et Yellow [24]. Un échantillon de huit problèmes réels, fournis par une commission scolaire, ont permis d'analyser les performances des heuristiques. Une méthode d'approche dans une implantation à grande échelle est

suggérée pour affecter les parcours aux points de transfert pour les étudiants du secondaire, et pour réduire le temps moyen qu'un étudiant passe dans l'autobus.

Dans les trois premiers chapitres, le lecteur est introduit à la problématique du transport scolaire et plus particulièrement à celle du transport scolaire en milieu rural. Les chapitres quatre à six décrivent les heuristiques utilisés dans notre implantation et le chapitre sept décrit une procédure d'optimisation de parcours. Le chapitre huit analyse les résultats obtenus à partir de problèmes reéls. Le chapitre neuf traite des problèmes rencontrés lors de l'implantation du système et propose une méthode d'approche. Le dernier chapitre conclue par une discussion des objectifs que nous avons atteints.

2. Méthodologie

Dans ce chapitre nous présentons la méthodologie employée pour traiter le problème de génération de parcours dans le transport scolaire. Cette description est abordée en deux temps : l'identification du problème, la recherche et l'élaboration des méthodes de solution.

2.1. Identification du problème

2.1.1. Définition d'un parcours

Un parcours est défini comme étant l'ordre dans lequel un véhicule parcourt un ensemble d'entités à servir. Pour le problème qui nous intéresse, une entité représente un segment de rue sur lequel réside au moins un étudiant voulant se rendre à l'école.

2.1.2. Différences entre le problème en milieu urbain et le problème en milieu rural

Dans le milieu urbain, étant donné la carrossabilité des routes et la forte densité de la population, les étudiants peuvent être rassemblés en des points que nous appellerons arrêts. Un parcours sera donc composé d'un certain nombre de ces arrêts, à partir desquels les étudiants seront transportés vers l'école.

Dans le milieu rural, il est difficile de rassembler les étudiants à un arrêt vue la faible densité de la population et l'état de certaines routes. Les étudiants sont donc

ramassés à domicile par l'autobus scolaire. Cette contrainte implique qu'un autobus doit parcourir chaque segment de rue sur lequel demeure au moins un étudiant.

2.1.3. Définition du problème

Notre problème se définit comme suit : Etant donné des segments de rue possédant au moins un étudiant voulant se rendre à l'école, générer des parcours respectant les contraintes de capacité d'autobus, de distance et de nombre maximum de segments à parcourir, tel que tous les étudiants soient transportés vers l'école et que la distance totale parcourue par les véhicules soit minimum.

2.1.4. Les points de transfert

Le point de transfert est utilisé comme un rassemblement régional des étudiants. Dans le milieu rural, les étudiants du secondaire, sont souvent rassemblés d'abord à un point de transfert, à partir duquel, ils sont ensuite acheminés vers leurs écoles respectives. Mais ceci ne s'applique pas aux étudiants du primaire, ceux-ci étant directement conduits vers leurs écoles respectives.

Dans ce qui suit, nous nous intéressons seulement à l'étape du ramassage des étudiants à leurs domiciles pour les acheminer vers les points de transfert ou les écoles. La deuxième étape lorsque nécessaire, pour acheminer les étudiants des points de transfert vers leurs écoles respectives, ne fait pas partie

de la présente étude car elle est intégrée au problème d'agencement de parcours dans le temps.

2.2. Recherche et élaboration des méthodes de solution

2.2.1. Transformation du problème

Notre problème consiste à parcourir des segments et à visiter des arrêts du réseau. Très peu d'algorithmes pour traiter ce genre de problèmes existent. Selon Orloff [21], le problème général de la génération de parcours sur les arrêts et segments peut se ramener à un problème plus simple sur les segments. Chaque arrêt i du réseau devant être traversé, peut être transformé en un segment artificiel (i,i) avec distance nulle.

2.2.2. Le réseau

Soit le réseau initial G (N,A) où N est l'ensemble des sommets du réseau et A est l'ensemble des segments du réseau, tel que $A \subset \{(i,j) / i,j \in N\}$.

Etant donné l'école E, soit $A' \subset A$ l'ensemble des segments dans A sur lesquels demeurent au moins un étudiant voulant se rendre à l'école E. De plus, A' contient le segment artificiel E − E représentant l'école. Soit $N' \subset N$ l'ensemble des sommets de N qui sont des extrémités des segments de A', $N' = \{i \in N / i$ sommet extrémité d'un segment de $A'\}$.

Pour construire le réseau réduit G (N',A'), on procède de la façon suivante : Pour chaque sommet de N', les chemins les plus courts pour se rendre à l'école E et à certains sommets de N' avoisinants, sont déterminés. Chaque segment de A' est ensuite considéré et un nombre fixé de liens artificiels (appelés arcs) sont créés pour relier les extrémités de ce segment aux extrémités d'autres segments appartenant à A'. Ces arcs correspondent aux chemins les plus courts déterminés précédemment pour les sommets extrémités de ce segment. Dans le cas particulier où un sommet de N' est l'extrémité d'au moins deux segments dans A', alors ce sommet est dédoublé et la distance de l'arc les reliant est nulle. Les figures 2.1, 2.2, et 2.3 illustrent les différentes étapes de la construction du réseau réduit G (N',A').

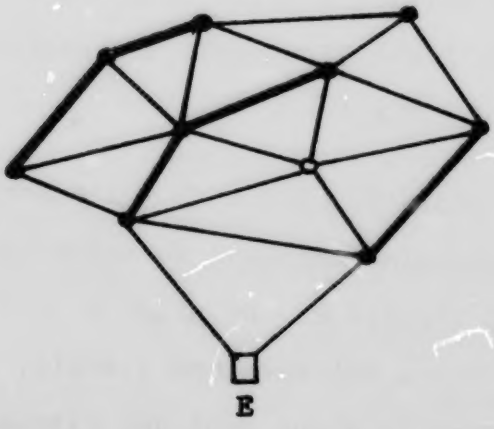


Fig. 2.1 Réseau initial G(N,A) avec les segments et les sommets à parcourir en noir foncé.

Fig. 2.2 Création des ensembles A', N'

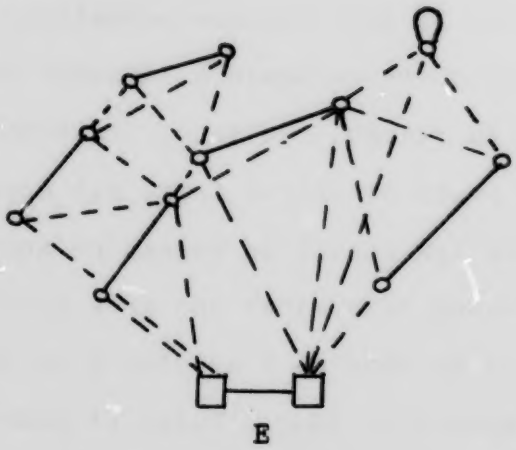


Fig. 2.3 Création du réseau réduit G(N',A')

Finalement, pour solutionner le problème, nous devons connaître la capacité des autobus utilisés, ainsi que la distance maximale et le nombre maximal de segments d'un parcours.

2.2.3. Approches pour la résolution du problème

Notre problème se définit donc comme un problème de génération de parcours sur des segments. Selon Orloff [22], ce problème sur les segments est soit appelé problème du "postier chinois", si tous les segments du réseau doivent être parcourus, ou bien problème du "postier rural", si seulement un sous-ensemble des segments du réseau doivent être parcourus. Une méthode exacte de solution peut être utilisée dans le cas où le réseau peut être parcouru par un seul véhicule, voir Christofides [5], Edmonds [10], Orloff [21]. Cette méthode se résume de la façon suivante. Si chaque noeud du réseau est de degré pair, alors le tour minimum permettant de traverser le réseau est de distance égale à la somme des distances de chaque segment du réseau. Ce tour, traversant chaque segment du réseau une seule et unique fois, est appelé cycle d'Euler. Si par contre les noeuds du réseau ne sont pas tous de degré pair, nous devons déterminer des arcs artificiels (constitués de segments du réseau) à ajouter à ce dernier pour garantir l'existence d'un tour d'Euler et pour minimiser la distance totale de ces arcs ajoutés. Une méthode exacte de solution, pour trouver ces arcs artificiels par "matching" des noeuds de degré impair du réseau, a été developpé par Edmonds.

Dans le cas où le réseau ne peut être parcouru par un seul véhicule, il est très difficile de trouver des méthodes de solution exactes parce que le problème est NP- complet.

Il faut alors se contenter d'un bon heuristique.

2.2.4 "Grouper en premier, circuler en second", "Circuler en premier, grouper en second".

Les principales méthodes de solution pour les problèmes de génération de parcours à m véhicules (problèmes de tournée) peuvent être classifiées sous l'une des grandes approches : "grouper en premier, circuler en second", et "circuler en premier, grouper en second".

L'approche "grouper en premier, circuler en second" consiste à diviser le territoire du réseau initial en sous territoires, chacun d'eux pouvant être parcouru par un seul véhicule respectant les contraintes de distance, de capacité et de nombre de segments. Les méthodes de Clarke & Wright [6], Insertion [15], et Yellow [24], toutes très utilisées dans la génération de parcours sur des arrêts peuvent être classifiées sous cette approche.

L'approche "circuler en premier, grouper en second" consiste à générer un grand tour comprenant toutes les entités à desservir en supposant infinie la capacité du véhicule et la distance maximale permise. Le grand tour est ensuite divisé en sous-parcours réalisable.

Notre approche au problème est celle de "grouper en premier, circuler en second", qui a prouvé être très efficace en milieu urbain. Les heuristiques proposés sont fortement

inspirés des heuristiques de Clarke & Wright, Insertion et Yellow tel qu'utilisé dans le problème de transport scolaire en milieu urbain (voir le texte de Forgues [12].

3. Etapes de la planification du transport scolaire

L'implantation d'un système informatisé de planification du transport scolaire comprend, comme nous l'avons déjà dit, plusieurs étapes. Dans une première étape, des parcours sont générés, et dans une deuxième étape ces parcours sont agencés dans le temps, afin de minimiser le nombre de véhicules utilisés, en faisant varier les horaires d'écoles (voir le texte de Desrosiers et al.[7]).

Le problème de génération de parcours est lui-même composé de plusieurs étapes. Le problème de génération de parcours en milieu urbain étant traité différemment de celui en milieu rural, ces deux problèmes sont solutionnés en parallèle (voir figure 3.1). En milieu urbain, les étudiants sont affectés à des arrêts et le problème du choix des arrêts se pose, (voir le texte de Dulac [8]). Des contraintes de temps de marche et de nombre maximum d'étudiants aux arrêts limitent l'affectation des étudiants aux arrêts. Pour ceci, si les parcours générés sur les arrêts ne sont pas jugés satisfaisants, ces contraintes sont redéfinis et les parcours sont générés à nouveau.

Dans le but de souligner les similarités et les différences de la génération de parcours, en milieu urbain (Forgues [12]) et en milieu rural, nous présentons un résumé des différentes contraintes rencontrées par les deux problèmes.

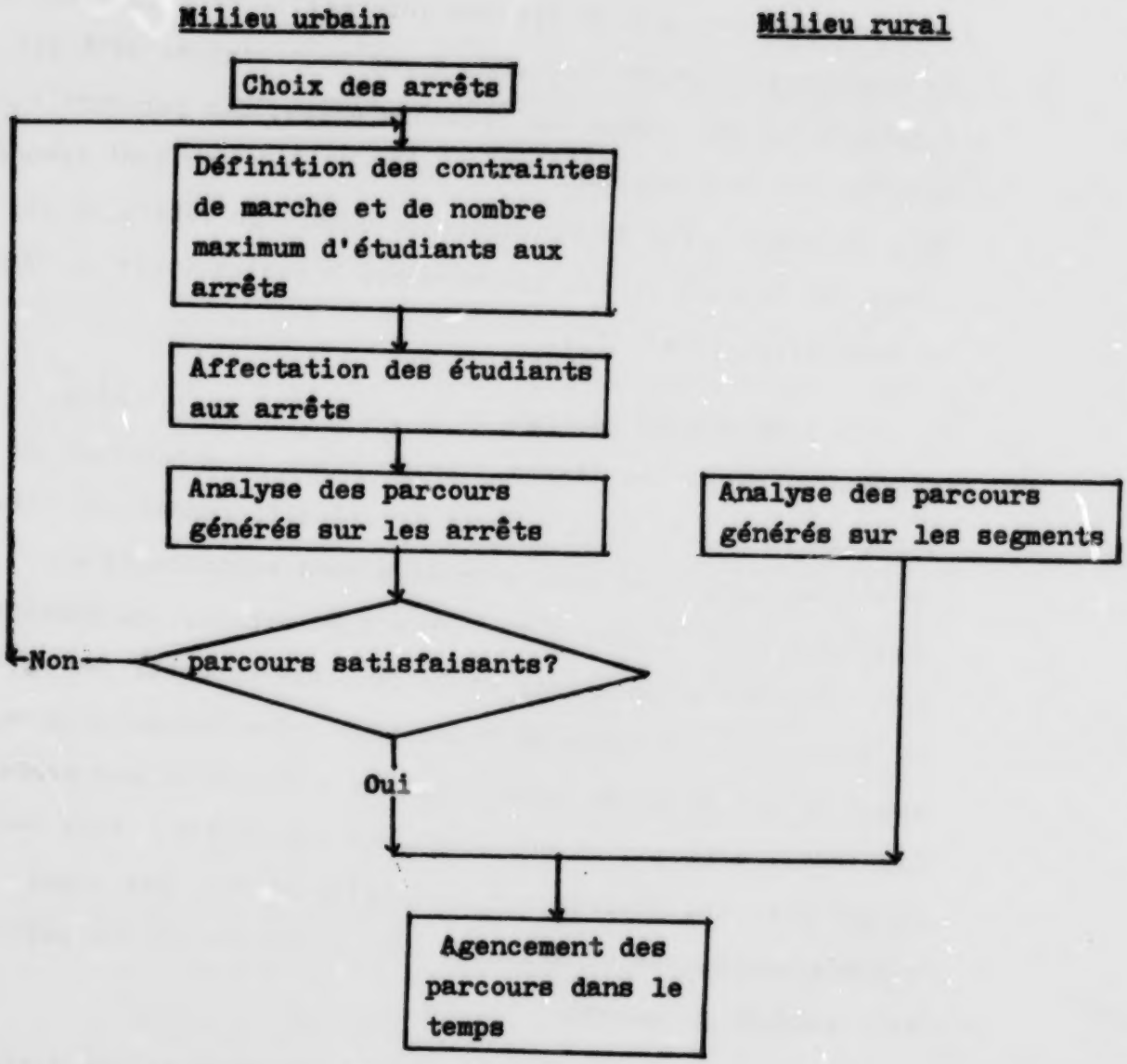


Figure 3.1. Etapes logiques de la génération de parcours en transport scolaire.

3.1. Génération de parcours sur les arrêts

Comme nous l'avons vu, la génération de parcours en milieu urbain s'effectue sur des arrêts. Une fois, le choix des arrêts et l'affectation des étudiants à ces arrêts terminés, il reste à générer des parcours sur ces arrêts.

Les contraintes sont perçues par les heuristiques comme des obstacles à la génération de parcours. Lors de la génération de parcours sur les arrêts, des contraintes sur le nombre d'arrêts, sur la distance et sur le nombre d'étudiants ramassés sont imposées.

La contrainte sur le nombre d'arrêts d'un parcours est violée si le nombre d'arrêts d'un parcours est plus grand que la limite fixée. Plus le nombre d'étudiants par arrêt est petit, plus la contrainte du nombre d'arrêts doit être grande.

La contrainte sur la distance est violée si la distance du parcours excède la limite fixée. Deux genres de parcours peuvent être générés. Pour les parcours sans origine à l'école, la distance est celle du trajet partant du premier arrêt et se rendant à l'école. Pour les parcours avec origine et destination à l'école, la distance est celle du trajet partant de l'école, visitant les arrêts et retournant à l'école.

La contrainte de capacité est respectée si un autobus de capacité suffisante peut être assigné au parcours. En

milieu urbain, la capacité des autobus est, en général, atteinte avant que les contraintes du nombre d'arrêts et de distance n'interviennent. Pour plus de détails le lecteur peut se référer au texte de Forgues [12].

3.2 Génération de parcours sur les segments

La génération de parcours en milieu rural s'effectue sur les segments. Ce problème est le sujet de la présente étude.

A chaque itération d'un heuristique, avant d'ajouter un segment au parcours, il faut vérifier qu'aucune des contraintes considérées n'est violée. Si une de ces dernières n'est pas respectée, le segment n'est pas ajouté au parcours. Les principales contraintes du problème sont maintenant décrites .

La contrainte sur le nombre maximum de segments à traverser dans un parcours n'a pas été liante pour la région étudiée. Nous l'avons, quand même, incluse dans notre implantation des heuristiques, au cas où cette contrainte jouerait un rôle pour d'autres régions.

La contrainte sur la distance maximum d'un parcours est très importante en milieu rural, étant donné la faible densité de la population. La distance du parcours est calculée à partir du sommet à l'extrémité du premier segment traversé jusqu'à l'école.

Dans le milieu rural, un seul type d'autobus est en général utilisé. Certains segments correspondent à des villages , par conséquent, ces derniers comportent souvent un nombre d'étudiants excédant la capacité d'un véhicule. Aussi, pour construire des parcours réalisables, ces segments devront être traversés autant de fois que la capacité des véhicules l'exigera. Il est à noter qu'il est parfois permis d'excéder légèrement la capacité d'un autobus, si cela lui permet de ramasser tous les étudiants d'un segment. Ceci se fait en proportion du taux d'absentéisme observé.

Une quatrième contrainte peut aussi être considérée. Elle combine les critères de distance et de capacité. Dépendemment des régions étudiées, il n'est parfois pas nécessaire de remplir à capacité l'autobus sur un parcours, cet espace libre étant par la suite utilisé dans l'acheminement des étudiants des points de transfert à l'école. Ainsi, une fois qu'un parcours comporte un certain nombre d'étudiants, il est parfois préférable de diriger l'autobus vers le point de transfert et de ne ramasser les étudiants sur un segment que si le détour n'est pas trop grand.

Etant donné la non uniformité de la distribution des étudiants sur les segments, il est difficile de prévoir, laquelle des deux contraintes, capacité d'autobus ou distance maximale, est la plus susceptible d'être violée. Le régisseur doit donc être conscient de ces particularités et plusieurs

alternatives devraient être envisagées avant de considérer une définition définitive des contraintes.

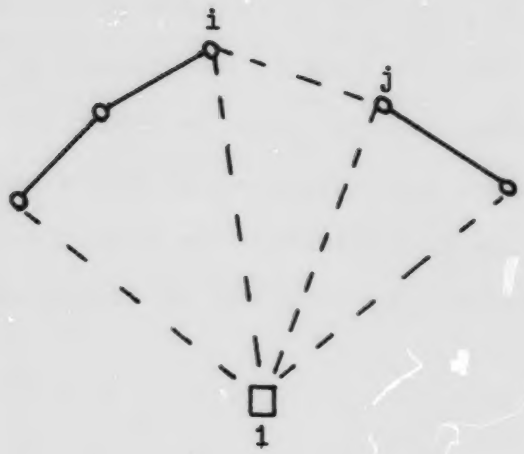
4. Méthode de Clarke & Wright

Cette section présente la méthode de Clarke & Wright, très couramment utilisée dans la génération de route pour le transport de marchandises et pour le transport scolaire en milieu urbain. Nous présentons aussi notre adaptation de la méthode au problème de génération de parcours sur des segments ou plus spécifiquement au problème de transport scolaire en milieu rural.

4.1. Introduction à la méthode

La logique de base de l'heuristique de Clarke & Wright, tel qu'utilisé dans la génération de parcours en milieu urbain est la suivante :

Créer initialement autant de parcours qu'il existe d'arrêts. Combiner ensuite les parcours itérativement de sorte que les nouveaux parcours respectent les contraintes de capacité d'autobus, de nombre d'arrêts et de distance. Le choix des couples de parcours à être fusionnés à chaque itération est fait en déterminant la combinaison qui implique la plus grande diminution de l'objectif, représentant le temps total de parcours des autobus. Par exemple, si les sommets extrêmes, i et j, de deux parcours sont reliés pour n'en former qu'un seul, alors l'épargne est évaluée comme suit :



$$(4.1) \qquad Sij = d(i,1) + d(j,1) - d(i,j)$$

L'épargne Sij est calculée pour chaque pair d'arrêts. Les épargnes sont triées en ordre décroissant, et le fusionnement des parcours est examiné dans l'ordre du tri. Un fusionnement s'effectue, seulement, si aucune des contraintes n'est violée.

Certains raffinements ont été ajoutés, par la suite, à l'heuristique de base de Clarke & Wright. Les plus importants sont les suivants :

1) utiliser un paramètre λ dans la formule de calcul des épargnes (4.1) permettant de choisir la meilleure structure de parcours par rapport au réseau. Cette transformation a été introduite par Yellow [24]. La formule d'épargne devient

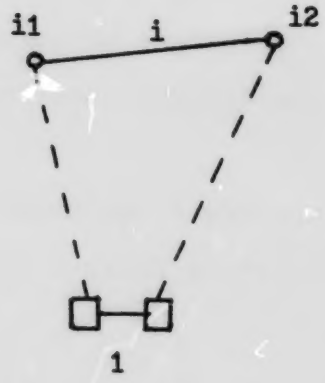
$$(4.2) \quad Sij = d(i,1) + d(j,1) - \lambda\, d(i,j);$$

2) trier les épargnes Sij à l'aide d'un "heap sort" pour réduire les opérations de comparaison et faciliter l'accès aux informations correspondantes (Voir Golden [16]);

3) considérer simplement les épargnes pour des sommets "rapprochés" les uns des autres.

4.2 Adaptation de Clarke & Wright au problème de routes sur les segments

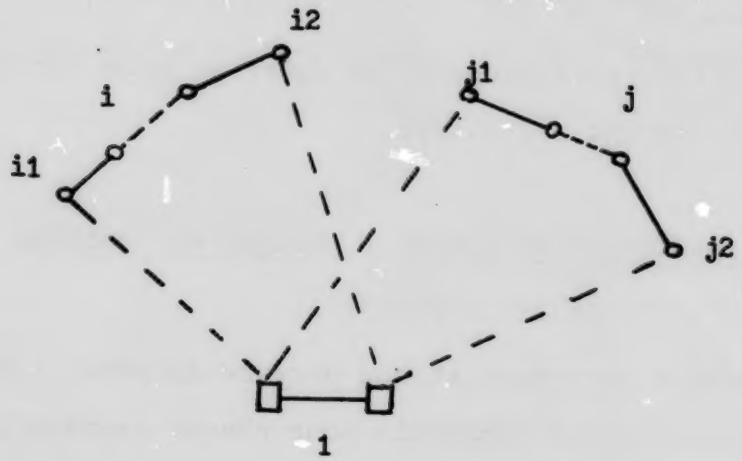
Le principe de l'heuristique demeure le même. Initialement un parcours est construit pour chaque segment i du réseau. La distance d'un parcours est calculée comme suit :



$$Di = d(i1,1) + d(i2,1) + d(i1,i2)$$

On essaye ensuite de réduire la distance totale parcourue

en fusionnant les parcours. Etant donné deux parcours i et j dont les sommets extrêmes reliés à l'école sont respectivement i1,i2 et j1,j2, quatre arcs existent permettant le fusionnement des deux parcours de la façon suivante :



L'arc (i2,j1) avec une épargne encourue Si2j1
L'arc (i2,j2) avec une épargne encourue Si2j2
L'arc (i1,j1) avec une épargne encourue Si1j1
L'arc (i1,j2) avec une épargne encourue Si1j2

Le choix des deux parcours à fusionner et la façon de les fusionner est fonction d'une épargne calculée par la formule suivante :

$$(4.3) \qquad S_{i.j.} = d(i.,1) + d(j.,1) - d(i.,j.).$$

La distance totale parcourue est réduite du facteur Si.j. , lorsque nous relions les sommets i. et j. situés respectivement aux extrémités des parcours i et j . La distance de l'école au sommet i. et au sommet j. est épargnée, mais la distance entre les sommets i. et j. , d(i.,j.), est encourue. Au début de la procédure les épargnes Si.j. sont calculées pour tous les arcs ai.j. , reliant les sommets i. et j. et elles sont triées en ordre décroissant à l'aide d'un "heap sort". Le fusionnement des parcours est examiné pour chaque arc, dans l'ordre du tri. Le fusionnement de deux parcours est sujet aux restrictions suivantes :

1) Les sommets i. et j. sont des sommets extrêmes du parcours.

2) Les segments incidents respectivement aux sommets i. et j. ne sont pas déjà dans le même parcours.

3) La contrainte sur la distance n'est pas violée.

4) La contrainte sur la capacité n'est pas violée. Cependant il y a un cas particulier à considérer lors de la vérification de la contrainte de capacité. Dans le cas où un des deux parcours à fusionner est composé de seulement un segment, le fusionnement s'effectue toujours, même si la contrainte sur la capacité est violée. Ainsi, le nouveau parcours comportera autant d'étudiants du segment que le

permet la capacité de l'autobus, les autres étudiants étant laissés pour un autre parcours.

4.2.1. Amélioration de la formule d'épargne

Afin d'obtenir une meilleure structure de parcours par rapport au réseau, le paramètre λ introduit par Yellow [27] est utilisé. La formule d'épargne (4.3) devient

$$(4.4) \quad Si.j. = d(i.,1) + d(j.,1) - \lambda\, d(i.,j.).$$

Ainsi plusieurs solutions à partir du même problème peuvent être générées en faisant varier le paramètre λ .

Dans le but d'améliorer les solutions obtenues par la méthode de Clarke & Wright, une formule pour le calcul des épargnes incluant un facteur de pénalité a été retenue. Ainsi, afin d'orienter les parcours vers l'école, la formule (4.5) est utilisée

$$(4.5) \quad Si.j. = d(i.,1) + d(j.,1) - \lambda d(i.,j.) + \left| d(i.,1) - d(j.,1) \right| .$$

Pour une discussion plus complète de la pénalité dans (4.5), le lecteur est referé au texte de Forgues [12].

Nous laissons le choix à l'usager d'utiliser l'une ou l'autre des formules (4.4) ou (4.5).

Les épargnes telles que calculées avec les formules (4.4) et (4.5) peuvent donner des Si.j. négatifs ou nuls. Notre implantation requérant des valeurs d'épargnes positives, nous additionnons une constante positive M pour nous assurer des Si.j. positifs. Les formules (4.4) et (4.5) deviennent respectivement

$$(4.4)' \quad Si.j. = d(i.,1) + d(j.,1) - \lambda\, d(i.,j.) + M$$

$$(4.5)' \quad Si.j. = d(i.,1) + d(j.,1) - \lambda\, d(i.,j.) + \left| d(i.,1) - d(j.,1) \right| + M$$

4.2.2. Particularité du problème sur les segments

En cas d'intersection de routes ou de carrefour, un sommet du réseau peut être l'extrémité de plus d'un segment du réseau.

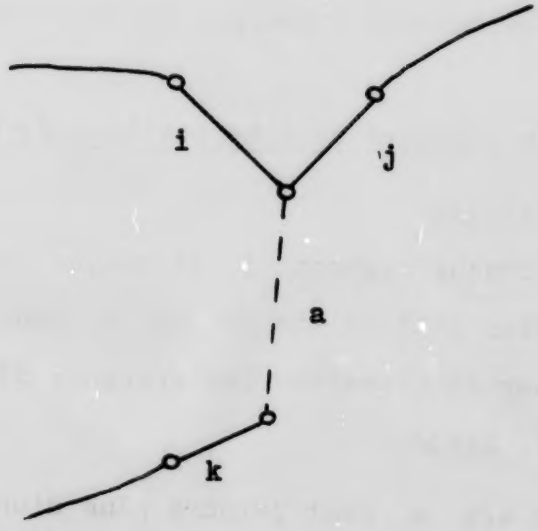


Figure 4.1

Un arc du réseau peut alors joindre plus d'une paire de segments. Par exemple dans la figure 4.1, l'arc a peut joindre les segments i et k ainsi que les segments j et k. Or, dans notre implantation, les épargnes sont associées aux arcs ajoutés pour former G(N',A'). Pour pallier à l'ambiguité dûe à des valeurs d'épargne qui, pour des fusions différentes, peuvent correspondre au même arc, la fusion avec la plus grande épargne correspondante est associée à l'arc. Dans l'exemple de la figure 4.1, si la meilleure épargne (associée à l'arc a) est encourue lors de la fusion d'un parcours incluant le segment i avec un autre parcours incluant le segment k , mais que cette fusion ne soit pas réalisable, il faut se rappeler que la fusion d'autres parcours incluant respectivement les segments j et k peut devenir intéressante éventuellement. Ceci est réalisé en associant à l'arc une autre fusion avec l'épargne correspondante.

4.3 Etapes logiques de l'heuristique de Clarke & Wright

Etapes	Description
[A]	Pour chaque segment i et chaque segment j , vérifier s'il existe un arc a pouvant les joindre. Evaluer et conserver les épargnes S_{i1j1}, S_{i1j2}, S_{i2j1}, S_{i2j2} .
[B]	Si un arc a peut joindre plus d'un couple de segments, garder pour l'arc a , la fusion permettant l'épargne la plus grande. Au cas où plusieurs

fusions correspondent à la même épargne,la première calculée sera la première considérée.

[C] Ordonner les épargnes en ordre décroissant.

[D] Retirer de la liste des épargnes, l'épargne Si.j. ayant la plus grande valeur.

[E] - Si Si.j. = 0 aller à [J] .

- Si le sommet i. ou j. n'est pas un sommet extrême d'un parcours, alors mettre Si.j. = 0 , aller à [I].
- Si les segments i et j sont dans le même parcours alors Si.j. = 0 , aller à [I].
- Si nombre de segments de parcours (i) + nombre de segments de parcours (j) > nombre maximum de segments alors Si.j. = 0 , aller à [I].

[F] Déterminer dans quel sens le nouveau parcours doit être traversé, pour qu'il soit orienté vers l'école. Evaluer la distance du nouveau parcours. Si la distance excède la distance maximale alors Si.j. = 0, aller à [I].

[G] - Si nombre d'étudiants du parcours (i) + nombre d'étudiants du parcours (j) $\leq$ capacité de l'autobus, aller à [H].
- Si parcours i ou parcours j composé de seulement un segment, alors dédoubler ce segment avec un nombre d'étudiants du parcours (i) + nombre d'étudiants du parcours (j) - capacité d'autobus, aller à [H]. Sinon Si.j. = 0 , aller à [I].

[H] Créer le nouveau parcours d'après la direction déterminée. Mettre à jour la distance, le nombre de segments, le nombre d'étudiants du nouveau parcours. Mettre Si.j. = 0.

[I] Vérifier si l'arc a = (i.,j.) peut relier un autre couple de segments. Si oui, aller à [C] si nécessaire, sinon aller à [D].

[J] Faire un OPT-2 (voir page 37) sur chaque parcours généré.

[K] Terminer.

5. Méthode d'insertion

Plusieurs heuristiques d'insertion existent pour la résolution de problèmes de commis voyageur. Les principales méthodes développées sont l'insertion la plus éloignée, la plus rapprochée, arbitraire et enfin celle du "convex hull".(Voir Golden [15].) Ces méthodes ont surtout été utilisées pour résoudre des problèmes de commis voyageur à un véhicule.

Une adaptation de l'heuristique d'insertion la plus éloignée pour un problème de commis voyageur à m véhicules a été utilisée par P. Forgues. Les résultats obtenus ont été jugés satisfaisants.

Cet heuristique d'insertion la plus éloignée a donc été adapté au problème du postier rural avec plusieurs véhicules ou, plus spécifiquement, au problème de transport scolaire en milieu rural. Nous nous sommes intéressés à la méthode d'insertion la plus éloignée parce qu'elle simule assez bien le travail manuel du régisseur dans la construction des parcours.

5.1 Adaptation de la méthode d'insertion au problème du transport scolaire en milieu rural

Le principe est le même que pour la méthode d'insertion pour le problème du commis voyageur à plusieurs véhicules (transport scolaire en milieu urbain). Il faut cependant travailler avec des segments et non plus seulement avec des arrêts.

Le principe est le suivant: Pour initialiser un parcours on sélectionne, parmi tous les segments hors parcours, celui qui est le plus éloigné de l'école, c'est-à-dire celui dont une des deux extrémités est le sommet le plus éloigné de l'école. Ce segment est alors considéré comme premier segment du parcours et sera traversé dans le sens du sommet le plus éloigné au sommet le moins éloigné de l'école. Partant de ce segment nous nous dirigeons vers l'école en essayant de ramasser le plus d'étudiants possible sur le chemin, c'est-à-dire d'insérer le plus de segments possible entre le segment initial et l'école.

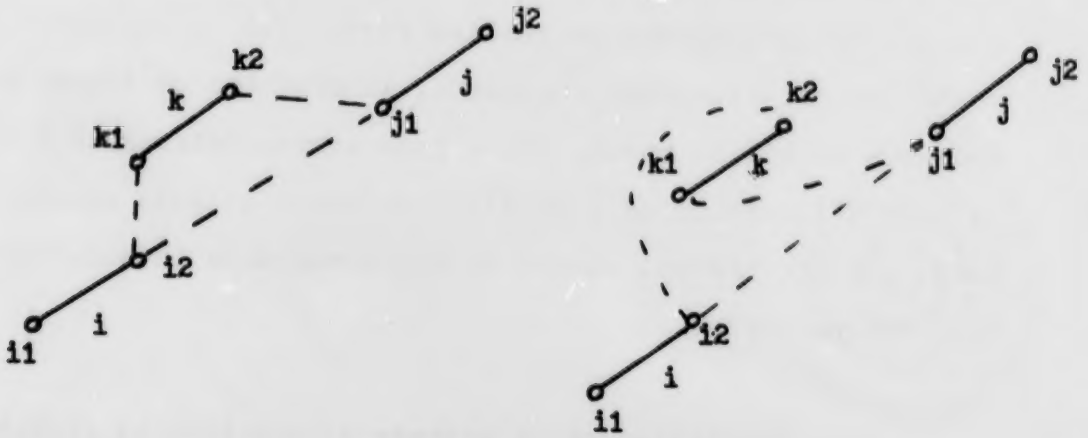


Figure 5.1

Etant donné les segments adjacents i et j du parcours, la direction de l'insertion éventuelle de chaque segment hors parcours k entre i et j est déterminée. Ceci est illustré dans la figure 5.1.

De même que pour le problème du commis voyageur, le choix du segment hors parcours à lui être intégré à chaque itération est effectué à l'aide d'un seul balayage de ces segments.

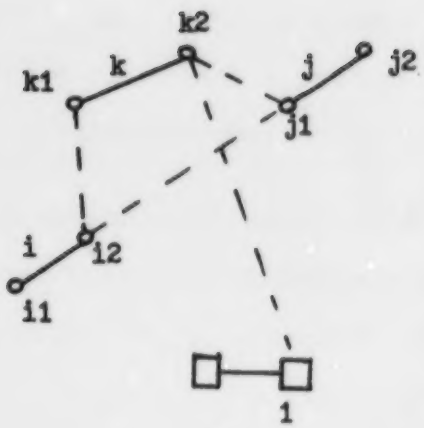
Avant d'insérer un segment k au parcours en construction, il faut vérifier la réalisibilité du nouveau parcours en terme de distance, de nombre de segments parcourus et de capacité de véhicule.

Si la contrainte de distance est violée, la présente insertion est irréalisable mais le segment k , peut dans certain cas être inséré ailleurs dans le parcours.

Si la contrainte de nombre de segments parcourus est violée, le parcours est complet.

Si la contrainte de capacité est violée, tous les étudiants du segment k , ne peuvent être ramassés par ce parcours. Le parcours en construction ramassera autant d'étudiants du segment k , que le permet la capacité du véhicule. Ce parcours sera alors saturé. Le segment k avec les étudiants qui restent sera considéré pour d'autres insertions ultérieures.

Dans notre implantation, pour chaque insertion réalisable du segment k entre les segments adjacents i et j du parcours en construction, la déformation est évaluée avec une des deux formules d'insertion, suivant que l'insertion est à l'extrémité du parcours ou non.



Soit $dikj = d(i2,k1) + d(k2,j1)$

$dk = d(k2,1)$

$dij = d(i2,j1)$

Alors la formule d'insertion, suivant le cas, s'énonce comme suit :

(5.1)

$$Cikj = -dij - dk + \lambda[dikj + d(k1,k2)]$$

V k segment hors parcours
i segment extrême du parcours en construction et $j = 1$.

$$Cikj = -dij - dk + \lambda[dikj + d(k1,k2)] + |d(i2,k1) - d(k2,j1)|$$

V k segment hors parcours
V i,j segments adjacents du parcours en construction ($j \neq 1$).

Si l'insertion du segment k n'est pas à l'extrémité du parcours, mais plutôt entre deux segments, i et $j \neq 1$, du parcours en construction P, alors une pénalité est introduite, afin qu'une insertion au milieu du parcours minimise

la déformation du parcours P. Ainsi le segment le plus "parallèle" au parcours P est introduit.

De tous les triplets d'insertion réalisable (i,k,j,), (i*,k*,j*), minimisant le coût de l'insertion, est retenu :

$$Ci^*k^*j^* = \min_{(i,k,j)} cikj \qquad \forall\ i,j \text{ adjacents} \in P \quad \forall\ k \text{ segment hors parcours}$$

5.2 Etapes logiques de l'heuristique d'insertion

Etapes	Description
[A]	Trouver parmi tous les segments hors parcours, le segment k dont l'une des extrémités k1 est la plus éloignée de l'école.
[B]	Former le parcours initial k - 1, le segment k étant parcouru de k1 à k2. Eliminer k de la liste des segments hors parcours. Si la liste des segments hors parcours est épuisé aller à [M].
[C]	Si les étudiants du segment k sont moins nombreux que la capacité de l'autobus aller à [E].
[D]	Remplir autobus jusqu'à capacité. Dédoubler segment k en segment k' tel que étudiants (k') = étudiants (k) - capacité de l'autobus. Enregistrer parcours. Aller à [A].
[E]	Pour chacun des segments k hors parcours, faire jusqu'à [G].

[F] - Si à l'itération précédante k représentait une insertion réalisable entre les segments ik et jk, alors vérifier si l'insertion de k demeure réalisable dans parcours augmenté.

- Si la contrainte du nombre de segments est violée aller à [K].
- Si la contrainte de distance est violée quand k est inséré entre ik et jk, alors annuler Ci_kkj_k et effectuer l'étape [G] pour tout couple (i,j) de segments adjacents dans parcours en construction.
- Si l'insertion demeure réalisable faire l'étape [G] pour les couples de segments adjacents (ik,jk), (i*,k*), (j*,k*).

[G] Parmi les couples de segments spécifiés, trouver le triplet (i,k,j) qui minimise la formule (5.1), et tel que l'insertion de k entre i et j soit réalisable.

[H] Soit le triplet (i*,k*,j*) tel que

$$Ci^*k^*j^* = \underset{(i,k,j)}{\text{Min}}\ Cikj \qquad \forall\ i,j \text{ adjacents dans } P$$

Si l'insertion de k* ne respecte pas la contrainte de distance, aller à [J].

Si l'insertion de k* ne respecte pas la contrainte de capacité aller à [I].

Insérer k* entre i* et j* dans parcours en

construction, retirer k* de la liste des segments hors parcours, aller à [E].

[I] Dédoubler segment k en segment k' , avec étudiants (k') = étudiants (k)-[capacité d'autobus-nombre d'étudiants dans parcours]. Insérer segment k dans parcours.

[J] Faire un OPT-2 (voir page 37) pour réduire distance du parcours. Si la construction du parcours avait été arrêtée à cause de la contrainte de distance pour au moins un k et si on n'a pas déjà exécuté un OPT-2 à l'itération précédente, aller à [H].

[K] Enregistrer parcours avec distance minimale. Aller à l'étape [A].

[L] Terminer.

6. Méthode de Yellow

Comme nous l'avons déjà dit, c'est Yellow qui a introduit le paramètre λ, dans la formule de base pour le calcul d'épargne de Clarke & Wright. Aussi, la méthode suggérée par lui, est une modification de la méthode de Clarke & Wright.

Théoriquement, l'heuristique de Clarke & Wright peut, de par sa conception, générer un grand nombre de parcours. Ceci vient du fait que plusieurs parcours sont construits en même temps. Ainsi il peut arriver que certains parcours ne peuvent être fusionnés sans violer une des contraintes, même si toutes les contraintes sont loin d'être liantes pour chacun d'eux. Ceci a pour effet d'augmenter le nombre de parcours.

La méthode de Yellow essaye de minimiser le nombre total de parcours, tout en minimisant la distance totale parcourue. Pour ceci, les segments ne sont ajoutés qu'à l'extrémité d'un parcours et un seul parcours est construit à la fois. Le choix des segments à ajouter à l'extrémité du parcours est fait en utilisant les mêmes formules d'épargne que précédemment (4.4)', (4.5)'. Quand aucun segment ne peut être ajouté sans violer une contrainte, le parcours est considéré final.

6.1 Etapes logiques de l'heuristique de Yellow

Etapes	Description
[A]	Pour chaque segment i et chaque segment j vérifier s'il existe un arc a pouvant les joindre. Evaluer et conserver les épargnes Si1j1, Si1j2, Si2j1, Si2j2.
[B]	Si un arc a peut joindre plus d'un couple de segments, garder pour l'arc a , la fusion permettant l'épargne la plus grande. Au cas où plusieurs fusions correspondent à la même épargne la première calculée sera la première considérée.
[C]	Ordonner les épargnes en ordre décroissant.
[D]	Retirer de la liste des épargnes, l'épargne Si.j. ayant la plus grande valeur.
[E]	- Si Si.j. = 0 , aller à [K]. - Si les segments i et j appartiennent à un parcours, alors Si.j. = 0, aller à [J]. - Si un des deux sommets, disons i., n'est pas une extrémité du parcours en construction, alors Si.j. = -Si.j., réordonner les arcs (i.,j.) en ordre décroissant de Si.j. , aller à [D].
[F]	Déterminer dans quel sens le nouveau parcours doit être traversé. Evaluer la distance du nouveau parcours. Si la distance excède la distance maximale, alors

Si.j. = 0 , aller à [J].

[F] Si nombre d'étudiants du parcours (i) + nombre d'étudiants du parcours (j) $\leq$ capacité d'autobus, aller à [H]. Sinon dédoubler le segment (j par exemple) devant être ajouté au parcours tel que étudiants de segment dédoublé j' = nombre d'étudiants du parcours (i) + nombre d'étudiants de j - capacité d'autobus.

[H] Créer le nouveau parcours d'après la direction déterminée. Mettre à jour la distance, le nombre de segments, la charge du nouveau parcours. Mettre Si.j. = 0.
Si la capacité de l'autobus a été atteinte aller à [I]. Sinon aller à [J].

[I] On a terminé de construire ce parcours, l'enregistrer de façon permanente.

- Si Si.j. $<$ 0 alors Si.j. = - Si.j.
- Ordonner tous les arcs en ordre décroissant de Si.j. , aller à [D].

[J] Vérifier si l'arc a = (i.,j.) peut relier un autre couple de segments. Si oui aller à [C]. Si nécessaire, sinon aller à [D].

[K] Faire un OPT-2 (voir page 37) sur chaque parcours généré.

[L] Terminer.

7. Algorithmes d'amélioration de parcours

Dans la description des heuristiques des sections précédentes, nous avons toujours inclus une étape d'optimisation de type OPT-2 des parcours générés. En effet, les heuristiques utilisés génèrent des parcours, mais ceci n'exclut pas la possibilité que certains parcours puissent être améliorés (c'est-à-dire, la distance parcourue réduite) en modifiant la séquence dans laquelle les segments sont traversés.

Cette technique a été surtout utilisée dans les problèmes de génération de parcours sur des arrêts. En effet dans le cas d'arrêts, le problème est équivalent à résoudre de façon heuristique un problème de commis voyageur à un véhicule. Lin [20] a développé des heuristiques de type OPT-2 et OPT-3 pour approximer la solution optimale du problème du commis voyageur. P. Forgues [12], a adapté ces heuristiques au problème du transport scolaire en milieu urbain. Le temps de solution de ces algorithmes est de complexité $O(n^k)$, où n est le nombre de sommets et k le degré d'optimalité OPT-k requis. Aussi dans l'implantation des heuristiques pour le transport scolaire en milieu urbain, les solutions obtenues par l'heuristique d'optimisation de type OPT-2 ont été jugées suffisantes. Nous l'avons adapté au problème de génération de parcours sur des segments, ou plus particulièrement au problème de transport scolaire

en milieu rural.

7.1. Adaptation de l'algorithme OPT-2

La rapidité d'exécution de la technique et le nombre réduit de segments dans les parcours rendent son application d'autant plus appropriée dans notre étude. L'algorithme OPT-2 utilise une solution initiale, en l'occurence un parcours, et le rend optimal de niveau 2 en procédant à des échanges d'arcs pour modifier la séquence dans laquelle les segments sont traversés.

En plus de l'examen des échanges de deux arcs contre deux autres arcs, Figure 7.1, nous examinons aussi le changement, essentiel dans le cas des segments, de la direction de traversée de certains segments. Figure 7.2, 7.3.

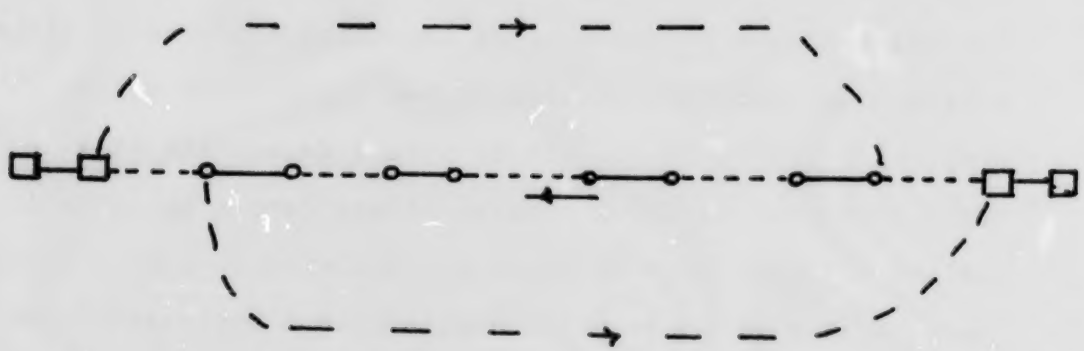


Figure 7.1

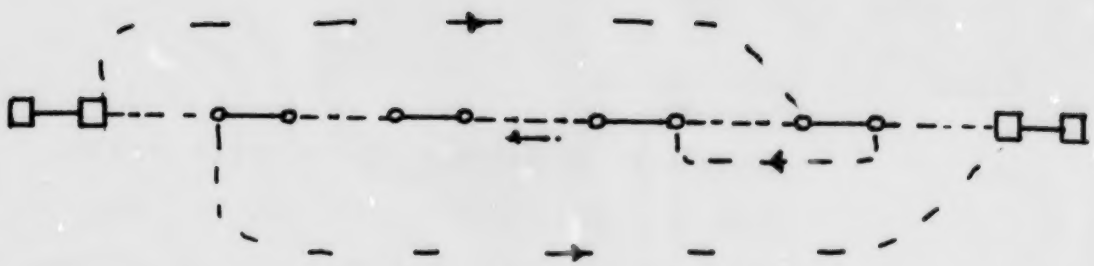


Figure 7.2

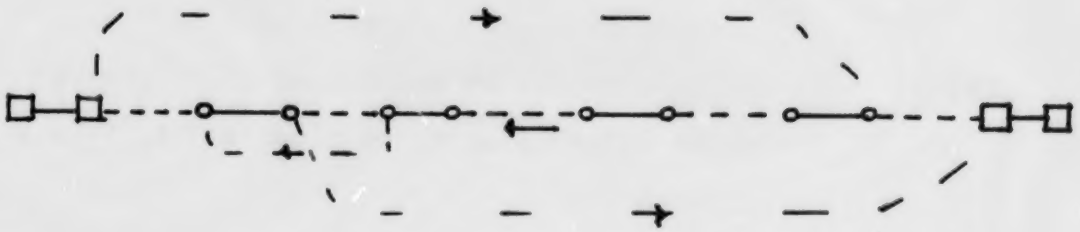


Figure 7.3

Si un de ces échanges réduit la distance, le parcours est ainsi modifié et l'examen des échanges recommencera.

La méthode ne garantit nullement l'optimalité globale mais elle garantit par contre, qu'aucun autre échange de deux arcs puisse réduire la distance du parcours.

7.2 Etapes logiques de l'adaptation de l'algorithme OPT-2

Etapes	Description
[A]	Créer dans le vecteur C le cycle initial sous la forme suivante C1, C2, ..., Cn, C1 (Ci contient le numéro du ième segment dans le cycle).
[B]	Faire jusqu'à [G] pour i = 1, ..., n - 2.
[C]	Faire jusqu'à [F] pour j = i + 1, ..., n - 1.
[D]	Examiner les trois types d'échanges tel qu'illustrés dans les figures 7.1, 7.2, 7.3.
[E]	Si on a trouvé un bon échange, l'effectuer comme illustré dans les figures 7.1, 7.2, 7.3. Aller à [B].
[F]	Itérer sur j.
[G]	Itérer sur i.
[H]	Calculer la nouvelle distance du parcours.
[I]	Terminer.

8. Analyse des résultats

Dans cette section nous présentons les résultats obtenus en appliquant ces heuristiques de génération de parcours sur des problèmes réels de transport scolaire. Nos résultats portent sur six réseaux différents, correspondants chacun, à une école primaire de la commission scolaire régionale St-François, couvrant le milieu rural de la région de Drummondville. Notre but est de comparer les performances des heuristiques développés sur différents réseaux.

8.1 Les problèmes, contraintes et heuristiques utilisés

Pour la comparaison des heuristiques nous avons choisi des réseaux de transport pour des écoles primaires de milieu rural, parce que, pour le primaire, le problème des points de transfert ne se pose pas, les étudiants étant directement conduits vers leur école. Nous reviendrons sur le problème des points de transfert dans le chapitre suivant. Le tableau 8.1 décrit les problèmes traités.

Les contraintes utilisées pour la génération de parcours sont uniformes pour chaque problème traité et nous ont été proposées par le régisseur de la commission scolaire. Un type d'autobus avec une capacité de 60 étudiants (mais

Problème	Taille du réseau			
	nbre segments	nbre sommets	nbre d'arcs	nbre d'étudiants
I	130	190	1445	253
II	115	148	1566	758
III	102	127	943	720
IV	87	110	854	644
V	77	83	626	497
VI	34	46	371	242

Tableau 8.1 Description des problèmes traités.

pouvant ramasser jusqu'à 63 étudiants, si nécessaire) est utilisé. Le nombre d'autobus utilisés, ainsi que le nombre de segments dans un parcours ne sont pas limités. La distance d'un parcours doit être inférieure à 45 kilomètres. Chaque extrémité d'un segment du réseau est reliée (par un arc) aux dix sommets les plus rapprochés dans le réseau. Les problèmes sont solutionnés avec tous les heuristiques selon le tableau 8.2.

8.2 Tableaux des solutions obtenues

Le tableau 8.3 résume les résultats. Il est à noter que cette expérimentation a été exécutée sur l'ordinateur CYBER 173 de l'Université de Montréal. A l'intersection heuristique - problème nous donnons la solution à distance minimale pour chaque nombre de parcours générés. Les temps de résolution en secondes sont donnés pour une exécution (sans inclure les temps d'entrées - sorties) de chacun des heuristiques utilisés.

Nous omettons la discussion du paramètre λ dans les formules (4.4)', (4.5)' et (5.1). Golden [15], [16] a en effet démontré qu'un examen de plusieurs λ doit être fait pour chaque problème à résoudre et qu'une valeur de λ générant des bons parcours pour un réseau pourrait, lorsqu'appliqué à un autre réseau, donner de mauvais résultats. Un balayage de plusieurs valeurs de λ doit, donc être fait

heuristique	formule	nombre de passes	valeurs de λ
Clarke & Wright (C W)	(4.4)'	11	λ= 1.5, 1.6,..., 2.4, 2.5
Clarke & Wright (C W)	(4.5)'	11	λ= 1.5, 1.6,..., 2.4, 2.5
Insertion (INSR)	(5.1)	3	λ= 0.9, 1.0, 1.1
Yellow (YLW)	(4.4)'	11	λ= 1.5, 1.6,..., 2.4, 2.5
Yellow (YLW)	(4.5)'	11	λ= 1.5, 1.6,..., 2.4, 2.5

Tableau 8.2 Description des heuristiques et paramètres utilisés.

Heuristique / Problème		C & W (4.4)'	C & W (4.5)'	INSR (5.1)	YLW (4.4)'	YLW (4.5)'
I	7p.		249323*			
	8p.	225055°				
	9p.		244797	247240		
	13p/14p.				296420 (14)	272464 (13)
	t	35.8	36.8	45.0	55.5	53.8
II	13p.		458578*			
	14p.		464737	464050°		
	15p.	464074		468101	525406	
	16p.					516307
		39.1	40.3	50.3	92.9	94.0
III	14p.		455411*	485311°	504834	
	15p.	470327		498204	504372	
	16p.				532973	
	t	23.4	23.7	31.8	42.5	

* = meilleur heuristique, t = temps C.P.U.

p = parcours o = second meilleur heuristique

Tableau 8.3 Sommaire des solutions heuristiques

Heuristique / Problème		C & W (4.4)'	C & W (4.5)'	INSR (5.1)	YLW (4.4)'	YLW (4.5)'
IV	13p.		452647°	447604*		
	14p.	424294		434240	479814	
	15p.	441924				470931
	t	19.3	19.9	28.4	34.7	34.5
V	9p.		256758°	261280*		
	10p.	261020				
	11p.				302059	301772
	t	17.2	17.4	22.8	27.4	27.7
VI	4p.	64470	63998*	64451°		
	5p.			64121		64678
	6p.				69329	
	t	3.1	3.8	5.0	5.9	5.9

Tableau 8.3 (suite)

pour chaque problème.

Etant donné que le nombre de parcours aussi bien que la distance totale parcourue varient d'un heuristique à l'autre, nous devons établir des critères afin de pouvoir évaluer la meilleure solution.

Ainsi, pour deux solutions comportant le même nombre de parcours celle minimisant la distance totale parcourue est choisie. Afin de pouvoir comparer des solutions n'ayant pas le même nombre de parcours, le critère retenu est basé sur le produit (nombre de parcours) x distance totale. La solution choisie est celle minimisant ce facteur. Le tableau 8.4 résume les solutions choisies suivant ce critère. On y remarque que Clarke & Wright a donné les "meilleures" solutions 4 fois sur 6 et que l'insertion vient en second avec 2 "meilleures" sur 6. En examinant le tableau 8.5, nous notons la nette dominance des heuristiques Clarke & Wright avec la formule (4.5)', et insertion sur l'heuristique de Yellow.

Etant donné que les heuristiques de Clarke & Wright et insertion donnent des résultats comparables il serait intéressant de souligner ici leurs particularités.

- L'heuristique de Clarke & Wright est plus rapide que l'insertion. Ceci est un avantage lors du choix d'un paramètre λ approprié au réseau étudié.

Problème	Heuristique	Nbre de parcours	Distance totale
I	C & W	7	249323
II	C & W	13	458578
III	C & W	14	455411
IV	INSR	13	447604
V	INSR	9	261280
VI	C & W	4	63998

Tableau 8.4 Tableau des solutions choisies.

Problème	Premier	Deuxième
I	C & W (4.5)'	C & W (4.4)'
II	C & W (4.5)'	INSR (5.1)
III	C & W (4.5)'	INSR (5.1)
IV	INSR (5.1)	C & W (4.5)'
V	INSR (5.1)	C & W (4.5)'
VI	C & W (4.5)'	INSR (5.1)

Tableau 8.5 Tableau des deux meilleurs heuristiques.

- L'heuristique de Clarke & Wright demande beaucoup plus d'espace mémoire que l'insertion. En effet, l'espace mémoire requis par Clarke & Wright est proportionnel au nombre d'arcs du réseau, tandis que l'espace mémoire requis par l'insertion est proportionnel au nombre de segments dans le réseau.

Dépendemment du problème traité, l'un ou l'autre des heuristiques, pourrait donc être utilisé.

9. Implantation de la génération de parcours en milieu rural

Dans ce chapitre nous discutons des différents problèmes que l'on pourrait rencontrer, lors de l'implantation d'un heuristique de génération de parcours en milieu rural. Nous suggérons également une méthode d'approche au problème dans une implantation à grande échelle.

9.1 Les points de transfert

Comme nous l'avons déjà mentionné le transport des étudiants du secondaire, en milieu rural, se fait en deux étapes. La première étape comprend le ramassage des étudiants vers un point de transfert. La deuxième étape comprend le transport des étudiants des points de transfert à leurs écoles respectives.

Pour chaque région il existe un certain nombre de points de transfert qui sont fixés et connus à l'avance. Ils sont, en général, localisés dans des villages aux alentours de la ville. Etant donné que les points de transfert peuvent être à promixité les uns des autres, l'assignation des étudiants aux différents points ne se fait pas, actuellement d'une façon systématique. Le seul critère suivi est le bon sens du régisseur et sa bonne connaissance des caractéristiques de sa région. Manuellement le régisseur procède de

la façon suivante : générer des parcours en considérant tous les étudiants et tous les segments du réseau. Quand un parcours est saturé, il est assigné au point de transfert le plus proche du dernier segment traversé et ceci tout en respectant les contraintes sur les points de transfert. Or étant donné que certains étudiants peuvent se rendre à pied à un point de transfert, il faut prévoir des places vacantes dans les autobus pour les transporter du point de transfert à l'école. Pour y arriver, quand un parcours non-saturé est jugé "assez rapproché" d'un point de transfert, alors ce parcours est considéré comme complet et assigné à ce point de transfert

L'assignation des parcours aux points de transfert est sujette à certaines contraintes, telle que des limites inférieures et supérieures sur la capacité des points de transfert, leurs situations géographiques par rapport à la ville et par rapport à la localisation des étudiants, etc.

Afin de déterminer une façon plus systématique de procéder, nous avons appliqué deux approches au problème.

9.1.1. Deux approches

La première approche que nous avons appliqué, est l'approche qui consiste à "circuler d'abord, diviser ensuite". Cette approche est semblable au travail qu'effectue le

régisseur manuellement. Tous les segments du réseau comportant des étudiants à transporter vers une école sont considérés en même temps pour la génération des parcours.

La deuxième approche est l'approche qui consiste à "diviser d'abord, ciculer ensuite". Dans cette approche une division du réseau en sous-territoires correspondants aux points de transfert est effectuée . Pour chacun des réseaux ainsi déterminé un heuristique de génération de parcours est alors appliqué.

Nous n'avons pas tenu compte, dans l'une ou l'autre approche, des contraintes quant aux points de transfert; le but de cet exercice est une comparaison "grosso modo" des deux approches qui nous permettra de déterminer laquelle est la plus appropriée pour une implantation à grande échelle.

9.1.2. "Circuler d'abord, diviser ensuite"

Etant donné tous les segments du réseau comportant des étudiants désirant se rendre à l'école, des parcours vers les points de transfert sont générés. Les parcours sont orientés vers un point du centre de la ville considéré comme l'école fictive. Les mêmes contraintes que précédemment régissent la construction des parcours. Quand un parcours est saturé, il est assigné au point de trans-

fert le plus proche, c'est-à-dire le point de transfert le plus rapproché du dernier sommet du parcours traversé.
Si un parcours en construction n'est pas encore saturé mais que son dernier sommet traversé est "assez proche" d'un point de transfert, alors ce parcours est considéré comme complété et assigné à ce point de transfert.

9.1.3. "Diviser d'abord, circuler ensuite"

Pour diviser le réseau en sous-réseau (chacun correspondant à un point de transfert),les segments comportant des étudiants , sont affectés aux points de transfert comme suit : Pour chacun des deux sommets d'un segment, touver le point de transfert le plus rapproché. Le segment est affecté au point de transfert le plus rapproché d'une des extrémités du segment. Un heuristique de génération de parcours est alors appliqué à chaque sous-réseau déterminé, le point de transfert correspondant étant considéré comme l'école vers laquelle sont dirigés les parcours.

9.1.4. Performance des deux approches

Afin de pouvoir comparer les deux approches nous avons appliqué un même heuristique de génération de parcours aux deux approches.

Nous avons choisi l'heuristique d'insertion à cause

de son besoin de mémoire réduit par rapport à celui requis pour Clarke & Wright. Il est ainsi bien adapté pour l'approche "circuler d'abord, diviser ensuite".

9.1.5. Résultats

Nous avons appliqué les deux approches au réseau de transport pour les écoles secondaires du milieu rural de la commission scolaire Régionale St-François. Le tableau 9.1 décrit l'affectation des segments aux septs points de transfert de ce réseau, tel que décrite dans la section 9.1.3. L'heuristique d'insertion a été exécuté pour chaque réseau généré. Il a été aussi exécuté avec le réseau initial.

Les résultats obtenus avec les deux approches sont très semblables, voir tableau 9.2. Le nombre total de parcours générés, ainsi que le nombre de parcours assignés à chaque point de transfert sont très semblables pour les deux approches. Donc, avec l'une ou l'autre des approches les résultats sont similaires. Il faudrait alors, essayer de les comparer en terme de facilité d'implantation.

- L'approche I, requiert beaucoup d'espace mémoire. Beaucoup d'artifices ont été utilisés pour réduire l'espace mémoire requis pour le réseau de la région étudiée. L'approche II, ne présente pas ces problèmes puisque l'heuristique est exécuté chaque fois sur un réseau réduit.

Point de transfert	Nombre de segments	Nombre de sommets	Nombre d'arcs	Nombre d'étudiants
A	76	82	620	497
B	86	109	860	640
C	12	15	200	130
D	23	26	254	529
E	111	147	1150	660
F	33	45	370	242
G	129	189	1200	257
TOTAL	470	613	4654	2955
Réseau initial	470	598	5073	2955

Tableau 9.1 Affectation des segments aux points de transfert pour l'approche II, "diviser d'abord, circuler ensuite".

	APPROCHE I "circuler d'abord, diviser ensuite"			APPROCHE II "Diviser d'abord, circuler ensuite"		
Point de transfert	Nombre d'étudiants	Distance parcourue (en Km)	Nombre de parcours	Nombre d'étudiants	Distance parcourue (en Km)	Nombre de parcours
A	421	273	8	497	261	9
B	456	334	10	644	461	14
C	120	44	2	130	39	3
D	730	193	13	529	89	9
E	681	490	16	660	512	15
F	324	112	6	242	65	6
G	223	192	6	253	230	7
TOTAL	2955	1638	61	2955	1657	63

Tableau 9.2 Comparaison des deux approches.

- Dans l'une ou l'autre approche, l'affectation des parcours aux points de transfert ne tient pas compte des contraintes relatives aux points de transfert. Il est beaucoup plus facile d'en tenir compte dans l'approche II, que dans l'approche I, étant donné que dans cette approche l'affectation des segments se fait à priori dans une étape indépendante de la génération de parcours.

- Etant donné que, dans l'approche II, chaque sous-réseau est traité séparément, il est possible de faire varier les paramètres de l'heuristique d'un réseau à l'autre pour répondre aux besoins de la région.

Il semble donc que l'approche II soit beaucoup plus appropriée pour la génération des parcours en milieu rural. Un algorithme pourrait être développé pour affecter les segments du réseau aux différents points de transfert, tout en tenant compte des contraintes relatives aux points de transfert.

Une procédure itérative pourrait aussi être développée. Une affectation initiale des segments aux points de transfert serait faite manuellement. L'heuristique de génération de parcours serait alors exécuté pour chaque sous-réseau généré. Après examen des résultats, le régisseur pourrait alors ajuster l'affectation et d'autres parcours seraient générés. Cette procédure serait répétée jusqu'à l'obtention de résultats satisfaisants.

9.2 Minimum global vs minimum local

Les heuristiques de génération de parcours que nous avons développé, cherchent à générer des parcours réalisables minimisant la distance totale parcourue. Il semble pourtant que, dans certaines régions, minimiser la distance totale parcourue ne soit pas le seul objectif de tout régisseur. En effet, il faut se rappeler que ce sont des étudiants que les véhicules transportent. Il est donc important de tenir compte du temps qu'un étudiant passe dans l'autobus puisque, en milieu rural, les distances parcourues sont assez grandes et puisque, dans certaines régions, la densité de la population est très faible.

Il est intéressant de voir pourquoi dans le milieu rural, les heuristiques minimisant la distance totale parcourue, ne minimisent pas forcément la distance de chaque parcours.

Figure 9.1

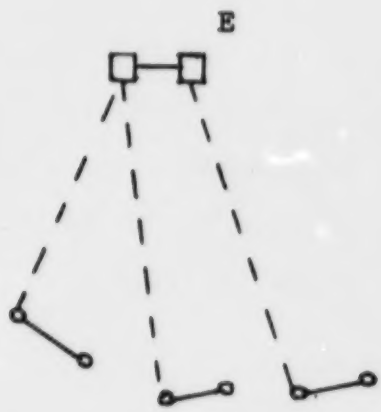


Figure 9.2

En effet, en essayant de minimiser la distance totale parcourue, on obtient des parcours de la forme de la figure 9.1. Si, par contre, on essayait de minimiser localement la distance de chaque parcours, on obtiendrait des parcours de la forme de la figure 9.2.

Il faut donc essayer de trouver un compromis, qui pourrait répondre aux besoins de chaque région.

9.2.1. Distance de ramassage

Une notion de distance de ramassage a été introduite. La construction d'un parcours se fait alors en deux étapes. Dans une première étape, tant que la distance de ramassage n'a pas été atteinte le parcours est construit comme précédemment en essayant de minimiser la distance totale, c'est-à-dire en essayant d'inclure dans le parcours le plus de segments dans le "voisinage" du parcours en construction.

Une fois que la distance parcourue a atteint la distance de ramassage, l'autobus est alors dirigé vers le point de transfert en ne ramassant que les étudiants sur sa route ou bien qui ne nécessitent qu'un détour négligeable de sa route initiale.

9.2.2. Adaptation de la méthode d'insertion

La méthode d'insertion, comme nous l'avons vu, permet une insertion réalisable d'un segment entre tout couple de segments adjacents du parcours en construction. Quand on essaye de minimiser la distance totale parcourue, ceci est plus efficace que de restreindre l'insertion aux extrémités (équivalent de la méthode de Yellow) comme en témoigne les résultats de la section 8.

Etant donné la logique de l'heuristique d'insertion, il est facile de l'adapter au concept de la distance de ramassage. Ainsi, tant que la distance de ramassage n'est pas atteinte, l'insertion d'un segment peut se faire comme auparavant entre n'importe quel couple de segments adjacents. Une fois la distance de ramassage atteinte, nous cherchons à diriger le parcours vers le point de transfert. Pour y arriver, seulement les insertions à l'extrémité du parcours (c'est-à-dire entre le dernier segment parcourue et le point de transfert),sont considérées.

9.2.3. Résultats

La valeur de la distance de ramassage est laissée à la discrétion de l'utilisateur. Si, par exemple, l'utilisateur désire minimiser la distance totale parcourue par les autobus et ne tient pas compte du temps moyen d'un étudiant

dans l'autobus, alors la distance de ramassage prend une valeur égale à la distance maximale d'un parcours. Si, par contre, l'utilisateur cherche à minimiser le temps moyen d'un étudiant dans l'autobus, alors la valeur de la distance de ramassage est ajustée en conséquence.

Afin d'illustrer l'effet de la variation de la valeur de la distance de ramassage nous l'avons implanté pour un point de transfert. Les résultats apparaissent dans le tableau 9.3.

Les résultats impliquent que la distance totale parcourue est au minimum avec une grande valeur de distance de ramassage. D'autre part, plus la distance de ramassage est petite et plus le temps moyen d'un étudiant dans l'autobus est petit.

Nous ne cherchons pas, ici, à définir une valeur optimale de distance de ramassage. Nous cherchons plutôt à montrer que la variation de la valeur de la distance de ramassage permet à l'utilisateur de générer des parcours répondants à ses besoins.

Distance de ramassage (en mètres)	Distance parcourue (en mètres)	Nombre de parcours	Kilomètrage d'un étudiant ds autobus (en mètres)	Ecart type du kilomètrage d'un étudiant dans un autobus
45000	507466	15	22193	7980
15000	543015	16	22346	7000
10000	557820	16	21176	5762
5000	550432	16	21008	5690
0*	539744	16	20779	5722

* Avec une distance de ramassage égale à 0, les segments sont ajoutés uniquement à l'extrémité d'un parcours.

Tableau 9.3 Variation de la distance de ramassage.

10. Conclusion

Lors de notre adaptation des heuristiques de génération de parcours au problème sur les segments et arrêts, nous avons remarqué que, pour les problèmes traités, la performance de ceux-ci est dépendante des réseaux sur lesquels ils sont appliqués. Mais d'une façon générale les performances des heuristiques de Clarke & Wright et insertion sont supérieures à celles de l'heuristique de Yellow pour le milieu rural.

Pour l'implantation d'un système de génération de parcours en milieu rural, l'affectation des étudiants du secondaire aux points de transfert dans une étape précédant la génération de parcours s'est avérée profitable.

Enfin, dans le but de réduire le temps moyen de l'étudiant dans l'autobus, la notion de distance variable de ramassage a été introduite. Une version modifiée de l'heuristique d'insertion incorporant cette notion permet au régisseur de pondérer l'objectif de minimisation de distance et l'objectif de réduire le temps moyen de l'étudiant dans l'autobus.

BIBLIOGRAPHIE

[1] Angel, R.D., Caudle, W.L., Noonan, R., Whinston, A.
Computer-assisted school bus scheduling
Management Science, 18 (1972), pp. 279-288.

[2] Beltrami, E.J., Bodin, L.D.
Networks and vehicle routing for municipal waste collection. Networks, 4 (1974), pp. 65-94.

[3] Bennet, B.T., Gazis, D.C.
School bus routing by computer. Transportation Research, 6 (1972), pp. 317-325.

[4] Bodin, L.D., Kursh, S.J.
A computer assisted system for the routing and scheduling of street sweepers. O.R.S.A., 26 (1978), pp. 525-537.

[5] Christofides, N.
The optimum traversal of a graph. Omega, 1 (1973), pp. 55-70.

[6] Clarke, G., Wright, J.W.
Scheduling of vehicles from a central depot to a number of delivery points. O.R.S.A., 12 (1964).

[7] Desrosiers, J., Ferland, J.A., Rousseau, J.M.
Horaires d'écoles et circuits d'autobus scolaire. Document de travail 88, Département d'informatique et de Recherche Opérationnelle, Université de Montréal, février 1976.

[8] Dulac, J.T.G.
Affectation des étudiants aux arrêts d'un système de transport des écoliers. Rapport de stage de Maîtrise, Département d'informatique et de Recherche Opérationnelle, Université de Montréal, Juin 1978.

[9] Edmonds, J.
Maximum matching and a polyhedron with 0,1 vertices. Journal of research of the national bureau of standards. 69 B (1965), pp. 125-130.

[10] Edmonds, J., Johnson, E.L.
Matching, Euler tours and the chinese postman problem. Mathematical programming,5 (1973), pp. 88-124.

[11] Ford, L., Fulkerson, D.
Flows in networks. Princeton University Press New Jersey (1962).

[12] Forgues, P.A.
Génération des parcours pour le transport scolaire, document de travail 101, Dept. d'Informatique et de Recherche Opérationnelle, Université de Montréal, Juin(1978).

[13] Fox, B.L.
Data structures and computer science techniques in operations research, publication 245, Dept. d'Informatique et de Recherche Opérationnelle, Université de Montréal, Décembre 1976.

[14] Gabow, H.N.
An efficient implememtation of Edmonds' algorithm for maximum matching on graphs. Journal for the ACM, 23 (1976), pp. 221-234.

[15] Golden, B., Bodin, L., Doyle, T., Stewart, W.Jr.
Approximate travelling salesman algorithms, Technical report 77-005, College of Business and Management, University of Maryland, May 1977.

[16] Golden, B.L., Magnanti, T.L., Nguyen, H.Q.
Implementing vehicle routing algorithms. Networks, 7 (1977), pp. 113-148.

[17] IBM Corp.
IBM vehicles scheduling program - extanded (USP-X) program description manual report SH 19 - 2304 - 0. White Plains N.Y.,(1968).

[18] Krolak, P., Felts, M., Nelson, J.
A man machine approach towards solving the generalized truck dispatching problem. Presented at the XXI AGARD meeting, Rome, May 1971.

[19] Kuon, Ea.
Une extension du problème du voyageur de commerce : Le problème de ramassage scolaire. Thèse présentée à l'Université scientifique et médicale de Grenoble pour obtenir le grade de Docteur de 3eme cycle en mathématiques appliquées.

[20] Lin, S.
Computer solutions of the travelling salesman problem. Bell system Technical Journal, 44 (1965), pp. 2245-2259.

[21] Orloff, C.S.
A fundamental problem in vehicle routing. Networks, 4 (1974(, pp. 35-64.

[22] Orloff, C.S.
Routing a fleet of M vehicles to / from a central facility. Networks, 4 (1974), pp. 147-162.

[23] Orloff, C.S., Caprera, D.
Reduction and solution of large scale vehicle routing problems. Transportation Science, 10 (1976), pp. 361-373.

[24] Yellow, P.C.
A computational modification to the savings method of vehicle scheduling, Operational Research Quarterly, 21 (1970), pp. 281-283.

ANNEXE I Liste sommaire des programmes utilisés

I - 1 Programme GNRFC

Génère le réseau réduit TAPE9 utilisé comme entrée au programme RTING

I - 2 Programme RTING

Génère les parcours d'autobus en milieu rural suivant l'heuristique désiré.

I - 1 Programme GNRFC

```
MNF(P=100000,R)

000000      PROGRAM GNRFC(INPUT,OUTPUT,TAPE1,TAPE7,TAPE9,
           +TAPE5=INPUT,TAPE6=OUTPUT)
      *
      ****************************************************
      *
      *  AUTEUR M. TANAGUI
      *
      ****************************************************
      *
      * FICHIERS UTILISES
      * ++++++++++++++++++
      * INPUT
      * TAPE1     RESEAU ORIGINAL DE LA COMMISSION SCOLAIRE
      * TAPE7     LISTE DES COMPRENANT DES ETUDIANTS DESTRANT SE RENDRE
      *           A L'ECOLE TRIEE PAR NUMERO DE SEGMENT
      *           (MUNICIPALITE,RUE,SEGMENT) ET LE NOMBRE D'ETUDIANTS
      *           ASSIGNES A CHAQUE SEGMENT
      *
      * OUTPUT
      * TAPE9     RESEAU REDUIT UTILISE POUR LA GENERATION
      *           DE PARCOURS
      *
      ****************************************************
      *
      * BUT DU PROGRAMME
      *
      ****************************************************
      *
      * ETANT DONNE LA LISTE DES SEGMENTS COMPRENANT DES
      * ETUDIANTS DESIRANT SE RENDRE A L'ECOLE,CE
      * PROGRAMME GENERE LE RESEAU REDUIT TAPE9 QUI SERA
      * UTILISE POUR LA GENERATION DES PARCOURS
      *
      ****************************************************
      *
      * DEFINITION DES VARIABLES GLOBALES
      * +++++++++++++++++++++++++++++++++++
      *
      * NBREN     NBRE DE SOMMETS
      * IH        NUMERO DU SOMMET
      * PARC      VECTEUR POINTANT AU PREMIER ARC ADJACENT
      *           A UN SOMMET
      * V         VECTEUR DE LA DISTANCE POUR SE RENDRE
      *           AU SOMMET I
      * ANCET     VECTEUR D'INDICE DU SOMMET PREDECESSEUR
      * VO        SOMMET ORIGINAL DANS L'ALGORITHME DES CALCULS
      *           DE DISTANCE
      * NO        NBRE MAXIMUM DE VOISINS A ATTEINDRE
      * CNT       COMPTEUR D'ITERATIONS
      * LH        POINTE A L'ORIGINE DE LA CHAINE
      * XI        COORDONNEE I DU SOMMET
      * XJ        COORDONNEE J DU SOMMET
      * ECOLE     NUMERO DE L'ECOLE
      * NBARC     NBRE D'ARCS DS RESEAU TAPE1
      * J         VECTEUR DE SOMMETS INITIALS DES ARCS
      * D         VECTEUR DE SOMMETS TERMINALS DES ARCS
```

```
*  P1        VECTEUR DE POINTEUR AU PROCHAIN ARC DE MEME ORIGINE
*  P2        VECTEUR DE POINTEUR AU PROCHAIN ARC DE MEME DESTINATION
*  DIST      VECTEUR DE LONGUEUR DES ARCS
*  VAL       DISTANCE POUR SE RENDRE A D(I) EN PASSANT PAR
*            L'ARC I
*  IORDRE    CHAINE DES ARCS POUR LE CHOIX MINIMUM
*  VOISIN    INDICATEUR DES VOISINS A CONSIDERER
*  FICME     INDICATIONS SUR LES SEGMENTS A PARCOURIR
*            1 SOMMET DEBUT DU SEGMENT
*            2 SOMMET FIN DU SEGMENT
*            3 NUMERO DU SEGMENT
*            4 IBRE D'ETUDIANTS
*            5 DISTANCE
*  IRC1      LISTE DES SOMMETS INCIDENT A DES SEGMENTS
*            TRIEE PAR NUMERO DE SOMMETS
*  NSEG      NBRE DE SEGMENTS
*  NARR      NBRE DE SOMMETS INCIDENTS A DES SEGMENTS
*
*
*  VARIABLES LOCALES
*  +++++++++++++++++
*  ITRV      LISTE DES SEGMENTS NON TROUVES DANS RESEAU
*
*
*  MODULES APPELES
*  +++++++++++++++
*  LIRE30       LIT LE RESEAU ORIGINAL TAPE1
*  TRV          RECHERCHE BINAIRE DU NUMERO DU SOMMET
*  MMDIST       CALCUL LES ND PLUS PROCHES VOISINS
*               AU SOMMET VU
*
*****************************************************************
      IMPLICIT INTEGER (A-Z)
```

```
000000        SUBROUTINE LIRESO
         *
         *****************************************************
         *
         * BUT DU MODULE
         *
         *****************************************************
         * CE MODULE LIT LE RESEAU ORIGINAL DE LA REGION TAPE1
         * IL VALIDE LES SEGMENTS A PARCOURIR PAR RAPPORT A LA
         * LISTE DES SEGMENTS DU RESEAU.IL IDENTIFIE LES SEGMENTS
         * DEVAIT ETRE PARCOURU
         *
         *****************************************************
         *
         * MODULES APPELES
         * +++++++++++++++
         * FIND      CHERCHE UN SEGMENT DANS LA TABLE DE SEGMENTS
         *           DU RESEAU
         * TRV       RECHERCHE BINAIRE DU NUMERO DU SOMMET
         *
         *****************************************************
000000        IMPLICIT INTEGER (A-Z)
```

```
0000008        SUBROUTINE FIND(SEG,K,IK)                                        1320
               *                                                                1321
               *****************************************************************1322
               *                                                                1323
               * BUT DU MODULE                                                  1324
               *                                                                1325
               *****************************************************************1326
               *                                                                1327
               * CE MODULE CERCHE UN SEGMENT DANS LA TABLE DE SEGMENTS          1328
               * DU RESEAU, SI CELUI CI N'EXISTE PAS IL EST AJOUTE OU           1329
               * REMPLACE PAR LE PLUS PROCHE SEGMENT DE LA TABLE DU             1330
               * RESEAU .SI LE NUMERO DU SEGMENT EST INVALIDE, CELUI CI         1331
               * EST ALORS REJETE                                               1332
               *                                                                1333
               *****************************************************************1334
0000030        IMPLICIT INTEGER (A-Z)                                           1335
```

```
000008          SUBROUTINE MINDIST(NB)                                          1539
                *                                                               1540
                ********************************************************        1541
                *                                                               1542
                * FIN DU MODULE                                                 1543
                *                                                               1544
                ********************************************************        1545
                *                                                               1546
                * CE MODULE CALCULE LES DISTANCES LES PLUS COURTES A            1547
                * PARTIR DU SOMMET VU AU PLUS PROCHES VOISINS                   1548
                * SUIVANT L'ALGORITHME DES CHEMINS LES PLUS COURTS              1549
                *                                                               1550
                ********************************************************        1551
000008          IMPLICIT INTEGER (A-Z)                                          1552
000008          COMMON/DONNEES/IM(01850),PARC(01850),V(01850),ANCET(01850),NBNEU,1553
```

```
000000B       SUBROUTINE PRDIST
        * ***************************************************************
        *
        * BUT DU MODULE
        *
        *************************************************************
        *
        * CE MODULE IMPRIME LES RESULTATS DU CALCUL DES CHEMINS LES
        * PLUS COURTS  .SI VO EST EGALE A L'ECOLE ALORS TOUS LES
        * SOMMETS DU RESEAU REDUIT SONT IMPRIMES ET LEUR DISTANCE A
        * L'ECOLE  . SINON LES NB SOMMETS SONT IMPRIMES AVEC LES
        * PLUS COURTES DISTANCES AU SOMMETS VO
        *
        ***********************************************************
0000..B       IMPLICIT INTEGER (A-Z)
```

```
000007B       SUBROUTINE TRV(NOEUD,IND)
          *
          ****************************************************************
          *
          * BUT DU MODULE
          *
          ****************************************************************
          *
          * CE MODULE EFFECTUE UNE RECHERCHE BINAIRE DU NUMERO DU
          * SOMMET DANS LA TABLE DES SOMMETS DU RESEAU.
          * SI LE SOMMET N'EXISTE PAS DANS LE RESEAU IND EST RETOURNE
          * AVEC LA VALEUR 0
          *
          ****************************************************************
000000B       IMPLICIT INTEGER (A-Z)
```

```
      PROGRAM RTING(INPUT,OUTPUT,TAPE9,TAPE5=INPUT,TAPE6=OUTPUT)
      IMPLICIT INTEGER(A-Z)
*******************************************************************
*
* AUTEUR : NADA TAWAGHI
*
*********************************************************************
*
* FICHIERS
*
*******************************************************************
* INPUT
* TAPE5       RESEAU DES ARRETS,SEGMENTS ET ARCS TEL QUE
*              GENERES PAR PROGRAMME GNRFC
* OUTPUT
* TAPE6       OUTPUT
*
********************************************************************
*
* DEFINITION DES VARIABLES GLOBALES
*
*********************************************************************
* DIMENSION EGALE AU NOMBRE DE SOMMETS
* ++++++++++++++++++++++++++++++++++++
*
* NOD(I)        POINTEUR AU DERNIER ARC OU SEGMENT
*               ADJACENT AU SOMMET I
* IARR(I)       LISTE DES NUMEROS DES SOMMETS I
* X(I)          COORDONNEES X DU SOMMET I
* Y(I)          COORDONNEES Y DU SOMMET I
*
*
* DIMENSION EGALE AU NOMBRE DE SEGMENTS
* ++++++++++++++++++++++++++++++++++++++
*
* ETU(K)        NOMBRE D'ETUDIANTS PAR SEGMENT K
* SEC(K)        POINTEUR AU SEGMENT K DANS
*                TABLEAU RESEAU
* CMRS(K)       CODE DE MUNICIPALITE,RUE,SEGMENT DU
*                SEGMENT K
* ROUTE(K,L)    INFORMATION CONCERNANT LE PARCOURS DU SEGMENT K
*                 ROUTE(K,1)=SEGMENT DEBUT DU PARCOURS
*                 ROUTE(K,2)=AUTOBUS UTILISE
*                 ROUTE(K,3)=NBRE D'ETUDIANTS DU PARCOURS
*                 ROUTE(K,4)=NBRE D'ARRETS DU PARCOURS
*                 ROUTE(K,5)=DISTANCEDU PARCOURS
* SEG(K)       PREDECCESSEUR DU SEGMENT K DANS LE PARCOURS
* SUCC(K)      SUCCESSEUR DU SEGMENT K DANS LE PARCOURS
* SOM       ESPACES DE TRAVAIL OU LES SEGMENTS
* SOL          INCLUS DANS UN PARCOURS SONT TEMPORAIREMENT MIS
*
* DIMENSION EGALE AU NOMBRE D'ARCS
* +++++++++++++++++++++++++++++++++
*
* S(L)         EPARGNES FAITES EN ASSOCIANT 2 SEGMENTS A L'AIDE
*               DE L'ARC L
* IN(L)        POINTEURS AVANT ET APRES
```

I - 2 Programme RTING

```
*  JH(L)        UTILISES DANS LE HEAP SORT
*  SEGI(L)       NOEUD DEBUT DE L'ARC
*  SEGJ(L)      NOEUD FIN DE L'ARC
*  SARC(L)       NBRE DE FOIS L'ARC L RELIE UN COUPLE DE SEGMENTS
*  DIMENSION EGALE NBRE DE SOMMETS+SEGMENTS+ARCS
*  +++++++++++++++++++++++++++++++++++++++++++++++++++
*
*  DIST         DISTANCE DANS LE RESEAU
*               DISTANCE SOMMETS A L'ECOLE
*               DISTANCE DES SEGMENTS
*               DISTANCE DES ARCS
*  ARCI         SOMMET DEBUT DES SEGMENTS OU ARCS DU RESEAU
*  ARCJ         SOMMET FIN DES SEGMENTS OU ARCS DU RESEAU
*  XI           POSITION DU PROCHAIN SEGMENT OU ARC
*                INCIDENT AU SOMMET ARCI
*  XJ           POSITION DU PROCHAIN SEGMENT OU ARC INCIDENT
*                AU SOMMET ARCJ
*
*
*  VARIABLES ET PARAMETRES
*  ++++++++++++++++++++++++++
*
*  NPT     NBRE DE SOMMETS DU RESEAU
*  NSEG    NBRE DE SEGMENTS DU RESEAU
*  NARC     NBRE D'ARCS DU RESEAU
*  SEGRES      NBRE DE SEGMENTS AUGMENTE DES SEGMENTS DEDOUBLES
*  NBR         NBRE DE PARCOURS GENERES
*
*  NFLOOP      INDIQUE SI LES PARCOURS DEBUTENT ET SE TERMINENT
*               A L'ECOLE OU BIEN DEBUTE AU PREMIER SEGMENT TRAVERSE
*  NDESS       INDIQUE SI UN GRAPHE DES PARCOURS GENERES
*               DOIT ETRE IMPRIME
*               1  OUI
*               0  NON
*  RDIST       DISTANCE DE RASSEMBLAGE UTILISE DANS HEURISTIQUE
*               INSERTION
*  NBUS        NBRE DE BUS UTILISES
*  MARKET       NBRE MAXIMUM DE SEGMENTS DS PARCOURS
*  MDIST          LONGUEUR D'UN PARCOURS
*  PAR        PARAMETRE DE CONSTRUCTION DE PARCOURS TEL
*              QU UTILISE DS LES FORMULES(CH 4,5,6)
*  NFALGO      NUMERO DE L'HEURISTIQUE CHOISI
*               1  INSERTION
*               2  CLARKEWRIGHT
*               3  YELLOW
*  NFINS      PARAMETRE INDIQUANT QUELLE FORMULE
*            D'EPARGNE CALCULER
*  NEOPT       INDIQUE SI LES PARCOURS DOIVENT ETRE OPTIMISE
*               A L'AIDE D'UN OPT2
*               1  OUI
*               0  NON
*  LISTE DES PRINCIPAUX MODULES UTILISES
*  ++++++++++++++++++++++++++++++++++++++++
*
*  LIGRAF    LECTURE DU RESEAU
*  RESULT    IMPRIME PARCOURS GENERES
```

PROGRAM RTING 73/173 OPT=1 FTN 4.6+460 15/07/79 17.55.0?

```
      * OPT2S        OPTIMIZE CHACUN DES PARCOURS GENERE
      *              A L'AIDE D'UN OPT2
      * STHEAP       TRI LE VECTEUR S( ) SOUS FORME
      *               DE "HEAP"
      * SAVING       CALCUL LES EPARGNES POUR CHACUN DES ARCS I,J,
      *                ET LES MET DANS VECTEUR S( )
      * CLKWT        HEURISTIQUE DE CLARKE&WRIGHT
      * INSRT        HEURISTIQUE D'INSERTION
      * YELLOW       HEURISTIQUE DE YELLOW
      *
      ********************************************************
            REAL T,T1
```

```
      C
            SUBROUTINE LIGRAF(FIN)
      *
      ************************************************************
      *
      *  BUT DU MODULE
      ************************************************************
      *
      * CE MODULE FAIT LA LECTURE DES DONNEES POUR LA
      * GENERATION DES PARCOURS.
      * IL ORGANISE LES DONNEES POUR PRESENTATION INTERNE DU
      * RESEAU .LES DONNEES SONT LUES DE LA FACON SUIVANTE
      *    DONNEES SUR L'ECOLE
      *    LES SOMMETS,LEUR DISTANCE A L'ECOLE ET LEURS
      *       COORDONNEES
      *    LES SEGMENTS AVEC LEURS SOMMETS DEBUT ET FIN ,LEURS
      *       NOMBRE D'ETUDIANTS ET LEURS DISTANCES
      *    LES ARCS RELIANTS LES SOMMETS ET LEURS DISTANCES
      *
      *
      * SI UNE ERREUR OU LA FIN DE FICHIER EST RENCONTREE
      * ON RETOURNE AVEC FIN=1 ;SI LA LECTURE S'EST BIEN
      * PASSEE ON RETOURNE AVEC FIN=0
      *
      ************************************************************
      *
      * MODULES APPELES ET LEUR BUT
      *
      ************************************************************
      *  RINR(L,NNA)     RECHERCHE BINAIRE DE NNA(I),I=1,..,L
      *                  ET REMPLACEMENT DE NNA(I) PAR SON INDEX
      *
      *    D(I,J)        FONCTION DONNANT LA DISTANCE DU
      *                  SOMMET I AU SOMMET J ,SI L'ARC N'EXISTE PAS
      *                  LA DISTANCE EST INFINIE
      *
      ************************************************************
            IMPLICIT INTEGER*2 (A-Z)
```

SUBROUTINE BINR 73/173 OPT=1 FTN 4.6+460 15/07/79 17.55.[illegible]

```
C
      SUBROUTINE BINR(L,NNA)
*
*****************************************************
*
* BUT DU MODULE
*****************************************************
*
* LE MODULE BINR FAIT UNE RECHERCHE BINAIRE DU
* NUMERO DU SOMMET NNA(I) I=1,...,L
* S'IL EXISTE UN K TEL IARR(K)=NNA(I) NOUS RETOURNONS AVEC
* NNA(I)=K . SINON NOUS RETOURNONS AVEC NNA(I)=-1 ET
* UN MESSAGE D'ERREUR EST IMPRIME.
*
* NOTONS QUE IARR(2),...,IARR(NPT) SONT EN ORDRE
* ASCENDANT MAIS NON IARR(1) QUI EST LE CODE DE L'ECOLE
*
*******************************************************
```

```
FUNCTION D          73/173  OPT=1                    FTN 4.6+460          15/07/79  17.55.06

      C
            FUNCTION D(M,N)
      *
      ****************************************************************
      *
      *  BUT DE LA FONCTION
      *
      ****************************************************************
      *
      * LE BUT DE LA FONCTION EST DE RETOURNER LA DISTANCE DE L'ARC
      * (M,N) DU RESEAU
      * SI L'ARC (M,N) N'EXISTE PAS LA VALEUR RETOURNEE EST
      * INFINIE
      * SI M EST EGAL A L'ECOLE ET LE PARAMETRE MFLOOP=0
      * ALORS D(1,N)=0 (LES PARCOURS NE COMMENCENT PAS A L'ECOLE)
      * SI M EST EGAL A N ALORS D(M,N)=0
      *
      ****************************************************************
            IMPLICIT INTEGER (A-Z)
```

```
      C
            SUBROUTINE RESULT
      *
      ************************************************************
      *
      *  BUT DU MODULE
      ************************************************
      *
      * CE MODULE EST APPELE PAR LE PROGRAMME PRINCIPAL POR
      *  POUR IMPRIMER LES PARCOURS GENERES
      * SI PARAMETRE MDESS=0,LE MODULE DESSINE LES
      * PARCOURS GENERES SUR IMPRIMANTE.
      * IL TRANSFORME LES COORDONNEES X,Y DES ARRETS
      * D'UN SEGMENT SUR UNE MATRICE 50X100, LES SEGMENTS
      * SONT INSERES A LEUR POSITION DS LE TABLEAU ET
      * SONT RELIES PAR DES POINTS.
      *
      * LE MODULE IMPRIME AUSSI DES STATISTIQUES PROPRES AUX
      * PARCOURS GENERES
      *
      ************************************************************
      *
      * DEFINITION DES VARIABLES LOCALES
      ************************************************************
      *
      * MINX        X( ) MINIMUM
      * MAXX        X( ) MAXIMUM
      * MINY        Y( ) MINIMUM
      * MAXY        Y( ) MAXIMUM
      * MX          ETENDUE DES X
      * MY          ETENDUE DES Y
      * IX          COORD X DANS LE TABLEAU
      * IY          COORD Y DANS LE TABLEAU
      * CARTE       TABLEAU 50X100
      * NOP         CARACTERE REPRESENTANT L'INDICE DU
      *               PARCOURS
      * DELTAX      DIFFERENCE ANCIEN NOUVEAU SEGMENT EN X
      * DELTAY      DIFFERENCE ANCIEN NOUVEAU SEGMENT EN Y
      * STEP        NBRE D'ETAPES ENTRE LES SEGMENTS
      * INCX        AUGMENTATION X PAR ETAPE
      * INCY        AUGMENTATION Y PAR ETAPE
      * VX          COORD X ENTRE LES SEGMENTS
      * VY          COORD Y ENTRE LES SEGMENTS
      * JX          ANCIENNE COORD X DS TABLEAU
      * JY          ANCIENNE COORD Y DS TABLEAU
      * STAT        STATISTIQUES SUR PARCOURS
      *
      ************************************************************
```

73/173 OPT=1 FTN 4.6+460 15/07/79 17.55.08

```
C
      SUBROUTINE OPT2S(N,SOL,SOM,COST)
*
*****************************************************************
*
* BUT DU MODULE
*
*****************************************************************
*
* CE MODULE A POUR BUT D'ARRANGER LES N SEGMENTS
* CONTENUS DANS LE VECTEUR SOM DE FACON QUE LE
* COUT DU PARCOURS DE CES SEGMENTS SOIT OPTIMAL
* DE NIVEAU 2.
* L'ARRANGEMENT FINAL EST DONNE DANS LE VECTEUR SOL
* ET LE COUT DANS COST.
*
* UN TOUR EST OPT2 SI ET SEULEMENT SI AUCUN
* ECHANGE DE TYPE OPT2 NE PEUT ETRE EFFECTUE SANS
* AUGMENTER LE COUT
*
*****************************************************************
*
* DEFINITION DES VARIABLES LOCALES
*++++++++++++++++++++++++++++++++++++++
*
*  N                   NBRE DE SEGMENTS
*  SOM(I),I=1,...,N    SEGMENTS DU PARCOURS A OPTIMIZER
*  SOL(I),I=1,...,N    SOLUTION DU TOUR MINIMAL OPT2
*  COST                COUT DU PARCOURS
*
*****************************************************************
*
* MODULES APPELES
* +++++++++++++
*
*  D(I,J)              DONNE LA DISTANCE ENTRE LES SOMMETS
*                        I ET J
*
*****************************************************************
      IMPLICIT INTEGER (A-Z)
```

```
  SUBROUTINE STRCAP      737173  OPT=1                    FTN 4.6+460          17/07/79  22.42.56

        C                                                                              5500
              SUBROUTINE STRCAP                                                        5510
        *                                                                              5511
        ****************************************************************               5520
        *                                                                              5521
        * BUT DU MODULE                                                                5530
        *                                                                              5531
        ****************************************************************               5540
        *                                                                              5541
        * CE MODULE QUI NOUS VIENT DE M.GOLDEN                                         5550
        * TRI LES EPARGNES S() EN ORDRE DESCENDANT DANS                                5560
        * STRCAP, IL CONSERVE LES POINTEURS AVANT-APRES                                5570
        * IM() ET JM()                                                                 5580
        *                                                                              5581
        ****************************************************************               5590
              IMPLICIT INTEGER (A-Z)                                                   5600
```

```
      SUBROUTINE OTHEAP(IC)                                                6000
C*************************************************************            6010
C*                                                                         6011
C*  BUT DU MODULE                                                          6020
C*                                                                         6021
C****************************************************************          6030
C*                                                                         6031
C*  CE MODULE NOUS VIENT DE B.GOLDEN                                       6040
C*  IL REARRANGE LE HEAP S() LORSQUE LA                                    6050
C*  VARIABLE S(IC) DOIT ETRE DESCENDUE DANS LE HEAP                        6060
C*  ET QU'ELLE DEVIENT PLUS PETITE QUE CELLES UTILISEES                    6070
C*                                                                         6071
C*************************************************************            6080
      IMPLICIT INTEGER (A-Z)                                               6090
      REAL S,PAR,COPY
```

```
     73/173  OPT=1                               FTN 4.6+460          17/07/79  22.42.51

      SUBROUTINE SAVING                                                        6370
                                                                               6371
*                                                                              6380
* **************************************************************               6381
*                                                                              6390
* BUT DU MODULE                                                                6391
*                                                                              6400
* **************************************************************               6401
*                                                                              6410
* LE MODULE CALCULE POUR CHAQUE ARC I-J L"EPARGNE S()                          6420
* REALISEES EN UNISSANT LES SOMMETS I-J SOUS UN PARCOURS.                      6430
*                                                                              6440
* NOUS AVONS RETENU DEUX METHODES DOMINANTES DU POINT                          6450
* DE VUE SOLUTION                                                              6460
* 1. LA METHODE ORIGINALE (MEINS=0)                                            6470
*    S(K)=D(I,1)+D(J,1)-PAR(D(I,J))+10000                                      6480
*                                                                              6490
* 2. LA METHODE AVEC PENALITE POUR MAUVAISE DIRECTION (MEINS=1)                6500
*    S(K)=D(I,1)+D(J,1)-PAR(D(I,J))+ABS(D(I,1)-D(J,1))+10000                   6510
*                                                                              6520
* NOUS ADDITIONNONS 10000 POUR QUE TOUS LES S(K) SOIENT POSITIFS               6530
*                                                                              6540
* LES EPARGNES S(K) POUR K=1,...,NN FORMANT UN HEAP QUI                        6550
* DEVRA ETRE ORDONNE PAR LA ROUTINE STHEAP                                     6560
* **************************************************************               6570
* LES CHAMPS ROUTE(,1),...,ROUTE(,5),PRED() ET SUCC()                          6580
* SONT INITIALISES DE FACON A REPRESENTER                                      6590
* UN PARCOURS PAR SEGMENT                                                      6600
* **************************************************************               6601
*                                                                              6602
* MODULES APPELES                                                              6603
* +++++++++++++++++                                                            6604
*                                                                              6605
* D(I,J)                  DISTANCE ENTRE SOMMETS I ET J                        6606
* TRVS(I,J,K,DIS,DIR)     CHERCHE L'ARC K RELIANT LES SOMMETS                  6607
*                         I ET J , DONNE LA DISTANCE DE L'ARC                  6608
*                         DIS ET SA DIRECTION DIR .SI L'ARC                    6609
*                         K N'EXISTE PAS DIS=INFNI                             6610
*                                                                              6611
* **************************************************************               6612
      IMPLICIT INTEGER (A-Z)
```

```
SUBROUTINE TRVS          73/173   OPT=1                         FTN 4.6+460          15/07/79  17.55.08

            SUBROUTINE TRVS(U,N,K,DIS,DIR)                                                      7810
      *                                                                                         7811
      ******************************************************                                    7812
      *                                                                                         7813
      * BUT DU MODULE                                                                           7814
      *                                                                                         7815
      *****************************************************                                     7816
      * CE MODULE CHERCHE L'ARC K RELIANT LES SOMMETS                                           7817
      * I ET J DU RESEAU , LA DISTANCE DE L'ARC K                                               7818
      * DIS ,AINSI QUE LE SENS DANS LEQUEL IL EST                                               7819
      * TRAVERSE ,DIR,SONT DONNES , SI L'ARC K N'EXISTE                                         7820
      * PAS DANS LE RESEAU ,DIS EST INFINIE.                                                    7821
      *                                                                                         7822
      *****************************************************                                     7823
```

```
                                    737173   OPT=1                           FTN 4.6+460        17/07/79  22.42.5

          SUBROUTINE CLKWT                                                                              A200
    *                                                                                                    A201
    ****************************************************                                                 A210
    *                                                                                                    A211
    * OUT DU MODULE                                                                                      A220
    *                                                                                                    A221
    ****************************************************                                                 A230
    *                                                                                                    A231
    * CE MODULE GENERE LES PARCOURS D"AUTOBUS SELON LA                                                   A240
    * METHODE DE CLARKE& WRIGHT TELLE QUE DECRITE                                                        A241
    * DANS LE CHAPITRE 4.                                                                                A242
    ****************************************************                                                 A260
    *                                                                                                    A261
    * DEFINITION DES VARIABLES LOCALES                                                                   A262
    * ++++++++++++++++++++++++++++++++                                                                   A263
    *                                                                                                    A264
    *  ETUCEK        INDIQUE SI TOUS LES ETUDIANTS D'UN                                                  A265
    *                  SEGMENT ONT ETE RAMASSES OU NON                                                   A266
    *                                                                                                    A267
    ****************************************************                                                 A268
    *                                                                                                    A269
    * MODULES APPELES                                                                                    A270
    * +++++++++++++++                                                                                    A271
    *                                                                                                    A272
    *   D(I,J)              CALCULE LA DISTANCE ENTRE LE SOMMET I ET J                                   A273
    *   UTHEAP(K)           REORGANISE LE "HEAP" QUAND S(K)=0                                            A274
    *   OLSAV((IS,JS,K)     VERIFIE POUR UN DONNE K SI D'AUTRES                                          A275
    *                       LIAISONS DE SEGMENTS PEUVENT ETRE                                            A276
    *                       ETABLIES AVEC L'ARC K                                                        A277
    *   TRVS(I,J,K,DIS,DIR)   CHERCHE L'ARC K RELIANT LES SOMMETS                                        A278
    *                       I ET J,LA DISTANCE DE L'ARC ,DIS ET                                          A279
    *                       SA DIRECTION SONT DONNES                                                     A280
    *                                                                                                    A281
    ****************************************************                                                 A282
          IMPLICIT INTEGER (A-Z)                                                                         A283
```

```
      *
            SUBROUTINE INSRT
      *
      ************************************************************
      *
      * BUT DU MODULE
      *
      ************************************************************
      *
      * CE MODULE GENERE LES PARCOURS D'AUTOBUS
      * SELON LA METHODE D'INSERTION TELL QUE DECRITE DANS
      * LE CHAPITRE 5.
      *
      ************************************************************
      *
      * DEFINITION DES VARIABLES LOCALES
      * ********************************
      *
      *  KHC(K)     CONSERVE LE SEGMENT I DU PARCOURS APRES
      *             LEQUEL K PEUT ETRE INSERE
      *  KHD(K)     CONSERVE LA VALEUR DE L'INSERTION DE K
      *  HLK        DONNE LA POSITION DES SEGMENTS HORS
      *             PARCOURS
      *
      ************************************************************
      *
      * MODULES APPELES
      * ***************
      *
      *  D(I,J)     DONNE LA DISTANCE ENTRE LES SOMMETS I ET J
      *  OPT2S      OPTIMISE CHAQUE PARCOURS GENERE A L'AIDE D'UN OPT2
      *
      ************************************************************
```

```
      SUBROUTINE YELLON                                                 17772
*                                                                       17773
*****************************************************************       17774
*                                                                       17775
* BUT DU MODULE                                                         17776
*                                                                       17777
*****************************************************************       17778
*                                                                       17779
* CE MODULE GENERE LES PARCOURS SELON L'HEURISTIQUE                     17780
* DE YELLON DECRIT DANS LE RAPPORT ,CHAPITRE 6.                         17781
*                                                                       17782
*****************************************************************       17783
*                                                                       17784
* DEFINITION DES VARIABLES LOCALES                                      17785
* --------------------------------                                      17786
*                                                                       17787
* ETUCEK      INDIQUE SI TOUS LES ETUDIANTS D'UN                        17788
*             SEGMENT ONT ETE RAMASSES OU NON                           17789
*                                                                       17790
*****************************************************************       17791
*                                                                       17792
* MODULES APPELES                                                       17793
* ---------------                                                       17794
*                                                                       17795
*  D(I,J)     DONNE LA DISTANCE ENTRE LES SOMMETS I ET J                 17796
*  STHEAP     ARRANGE LE "HEAP" UNE FOIS QUE TOUTES LES                  17797
*             EPARGNES ONT ETE CONSIDERES                               17798
*  OTHEAP     REORGANISE LE "HEAP" QUAND S(K) S'ANNULE                   17799
*             UNE VALEUR DE K                                           17800
*  RESAV      VERIFIE POUR UN ARC DONNE K SI D'AUTRES                    17801
*             LIAISONS DE SEGMENTS PEUVENT ETRE ETABLIES AVEC           17802
*             L'ARC K                                                   17803
*                                                                       17804
*****************************************************************       17805
      IMPLICIT INTEGER (A-Z)                                            17806
```

```
      SUBROUTINE RESAV(IS,JS,K)
*
**************************************************
*
* BUT DU MODULE
*
**************************************************
*
* ETANT DONNE L'ARC K ,LE MODULE VERIFIE S'IL EXISTE
* D'AUTRES COUPLES (I,J) DE SEGMENTS POUVANTS ETRE
* LIES A L'ECOLE A L'AIDE DE K, QUE CEUX DEJA CONSIDERES
*
****************************************************
*
* DEFINITION DES VARIABLES LOCALES
* +++++++++++++++++++++++++++++++++
*
*  KEEP(K)       VALEUR DE L'EPARGNE POUR L'ARC K
*  KEEPI(K)      SEGMENT I
*  KEEPJ(K)      SEGMENT J
*
*****************************************************
*
* MODULES APPELES
* +++++++++++++++
*
*  D(I,J)    DONNE LA DISTANCE ENTRE LES SOMMETS I ET J
*
*****************************************************
```

```
SUBROUTINE OPTROUT     73/173  OPT=1                    FTN 4.6+460          15/07/79  17.55.08

            SUBROUTINE OPTROUT                                                      22240
      *****************************************************************             22241
      *                                                                             22242
      * BUT DU MODULE                                                               22243
      *                                                                             22244
      * ******************************************************************          22245
      *                                                                             22246
      * CE MODULE EST APPELE APRES L'EXECUTION DE CLARKE &WRIGHT                    22247
      * OU YELLOW  IL PREPARE LES PARCOURS GENERES PAR CES HEURISTIQUES             22248
      * A UNE OPTIMISATION DE TYPE OPT2                                             22249
      *                                                                             22250
      ***********************************************************************       22251
      *                                                                             22252
      * MODULES APPELES                                                             22253
      * +++++++++++++++                                                             22254
      *                                                                             22255
      * OPT2C   OPTIMISE CHAQUE PARCOURS GENERE A L'AIDE D'UN                       22256
      *         OPT2                                                                22257
      *                                                                             22258
      ******************************************************************            22259
      .                                                                             22260
```